



**Universitat de Lleida
Escola Politècnica Superior
Enginyeria Tècnica en Informàtica de Gestió**

Trabajo Final de Carrera

**DISEÑO E IMPLEMENTACIÓN DE UNA APLICACIÓN
DE GESTIÓN PARA UNA EMPRESA EDITORIAL**

Directora: Marga Moltó Aribau

Autor: Sergio Grau Vilas

Septiembre 2007

Índice

1. Introducción.....	6
2. Objetivos.....	7
3. Modelo de Negocio	8
3.1. Análisis del modelo	8
4. Método de desarrollo	10
4.1. Introducción.....	10
4.2. Modelos de desarrollo de software	12
4.2.1. Modelo secuencial o en cascada.....	12
4.2.2. Modelo de prototipos o modelo de desarrollo evolutivo	13
4.2.3. Modelo Incremental.....	14
4.2.4. Modelo en espiral	15
4.2.5. Conclusiones.....	16
4.3. Proceso de desarrollo del software	17
5. Análisis de requisitos y especificación del sistema	19
5.1. Requisitos funcionales.....	19
5.1.1. Modelo de Casos de Uso	20
5.2. Requisitos no funcionales.....	34
5.3. Especificaciones del sistema	35
5.3.1. Hardware	35
5.3.2. Software.....	35
5.4. Necesidades del usuario.....	36
5.4.1. Perfil del usuario.....	36
5.4.2. Las funcionalidades de la aplicación	36
5.5. Análisis de la competencia	37
5.6. Prototipos de papel	37
6. Planificación y análisis del riesgo	38
6.1. Identificación y gestión del riesgo.....	38
6.2. Planificación	40
6.3. Presupuesto de proyecto	41
7. Base de datos	43
7.1. Modelo Entidad-Relación.....	43
7.2. Modelo Relacional.....	44
7.3. Diccionario de datos	46
8. Diseño del sistema	50
8.1. Análisis jerárquico de tareas (HTA).....	50
8.2. Diseño externo.....	56
8.2.1. Modelo de la interfaz.....	56
9. Implementación	59
9.1. Herramientas de desarrollo	59
9.1.1. ¿Qué es la plataforma .NET?.....	59
9.1.2. Visual Basic 2005	59
9.1.3. ¿Por qué Visual Basic 2005?	63
9.1.4. Crystal Reports	63
9.2. Un poco de código.....	64
9.2.1. Procedimientos Almacenados.....	64
9.2.2. Visual Basic	69

10. Pruebas y validación	72
10.1. Pruebas de unidad	72
10.2. Pruebas de integración.....	73
10.3 Pruebas de validación	73
10.4 Pruebas de sistema.....	74
10.4.1. Prueba de recuperación.....	74
10.4.2. Prueba de seguridad.....	74
10.4.3. Prueba de resistencia	74
10.4.4. Prueba de rendimiento	75
11. Conclusiones y ampliaciones	76
11.1 Conclusiones.....	76
11.2. Ampliaciones	76
12. Bibliografía.....	77
ANEXO I: Manual de usuario	78
ANEXO II: Clase Basedatos	119

Índice de Figuras

Figura 1: Modelo secuencial o en cascada	12
Figura 2: Construcción de prototipos	13
Figura 3: Modelo incremental	14
Figura 4: Modelo en espiral	15
Figura 5: Caso de Uso “Autenticación”	21
Figura 6: Caso de Uso “Alta Autor”	22
Figura 7: Caso de Uso “Modificar Autor”	23
Figura 8: Caso de Uso “Eliminar Autor”	24
Figura 9: Caso de Uso “Alta Sello”	25
Figura 10: Caso de Uso “Modificar Sello”	26
Figura 11: Caso de Uso “Eliminar Sello”	27
Figura 12: Caso de Uso “Alta Libro”	28
Figura 13: Caso de Uso “Modificar los datos del libro”	29
Figura 14: Caso de Uso “Eliminar un libro”	30
Figura 15: Caso de Uso “Movimiento de libros”	31
Figura 16: Caso de Uso “Venta de libros”	32
Figura 17: Caso de Uso “Facturar a un Distribuidor”	33
Figura 18: Modelo Entidad-Relación	43
Figura 19: Diagrama de la base de datos	44
Figura 20: Tarea “Alta Libro”	51
Figura 21: Tarea “Modificar Libro”	52
Figura 22: Tarea “Eliminar un libro”	53
Figura 23: Tarea “Cargar Ejemplares”	54
Figura 24: Tarea “Movimiento entre distribuidores”	55
Figura 25: barra de herramientas de la aplicación	56
Figura 26: Diseño de las pantallas de Gestión	58
Figura 27: Ejemplo de Tipos de datos parciales	61
Figura 28: Archivo XML con los comentarios del proyecto	62
Figura 29: Diagrama de clases “Herencia visual”	70
Figura 30: Diseño de formulario “frmListadoBase”	70

Índice de tablas

Tabla 1: Flujo de acontecimientos “Autenticación”	21
Tabla 2: Flujo de acontecimientos “Alta Autor”	22
Tabla 3: Flujo de acontecimientos “Modificar Autor”	23
Tabla 4: Flujo de acontecimientos “Eliminar Autor”	24
Tabla 5: Flujo de acontecimientos “Alta Sello”	25
Tabla 6: Flujo de acontecimientos “Modificar Sello”	26
Tabla 7: Flujo de acontecimientos “Eliminar Sello”	27
Tabla 8: Flujo de acontecimientos “Alta Libro”	28
Tabla 9: Flujo de acontecimientos “Modificar los datos del libro”	29
Tabla 10: Flujo de acontecimientos “Eliminar un libro”	30
Tabla 11: Flujo de acontecimientos “Movimiento de libros”	31
Tabla 12: Flujo de acontecimientos “Venta de libros”	32
Tabla 13: Tareas del proyecto	40
Tabla 14: Presupuesto material fungible	41
Tabla 15: Presupuesto.....	42
Tabla 16: Tabla “Sellos” de la base de datos.....	46
Tabla 17: Tabla “Estilos” de la base de datos	46
Tabla 18: Tabla “Colecciones” de la base de datos.....	46
Tabla 19: Tabla “Autores” de la base de datos.....	46
Tabla 20: Tabla “Libros” de la base de datos.....	47
Tabla 21: Tabla “DistribuidoresClientes” de la base de datos	48
Tabla 22: Tabla “Almacenes” de la base de datos.....	48
Tabla 23: Tabla “FormasDePago” de la base de datos.....	48
Tabla 24: Tabla “Usuarios” de la base de datos	48
Tabla 25: Tabla “Movimientos” de la base de datos	49
Tabla 26: Tabla “Facturas” de la base de datos.....	49
Tabla 27: Tabla “FacturasDetalles” de la base de datos.....	49

1. Introducción

La aplicación que he desarrollado, surge a partir de observar la necesidad de gestionar, mediante un sistema informático, la actividad empresarial de una pequeña o mediana editorial. Algunas editoriales tienen que recurrir a varias aplicaciones, para gestionar sus procesos. Por ejemplo, para llevar el control de los stocks de cada tienda, la editorial para la que se ha realizado este proyecto, utilizaba una hoja de cálculo.

GESTEDI se ha desarrollado para satisfacer las necesidades de un cliente concreto, Aun así, no se descarta la comercialización de esta aplicación.

A la hora de diseñar esta aplicación, se ha intentado hacerla lo mas manejable y usable posible, para que se pueda utilizar de forma sencilla, práctica y útil. Evitando, por ejemplo, la sobrecarga de botones en los formularios.

Esta memoria consta de los siguientes capítulos:

- Objetivos que persigue la aplicación.
- Descripción del método de desarrollo utilizado para realizar la aplicación.
- El modelo de negocio, una amplia descripción del tipo de empresa a informatizar.
- Análisis de requisitos y especificación del sistema, donde se describe su estructura y funcionamiento así como las características físicas del entorno donde se instalará.
- Planificación del proyecto y análisis del riesgo.
- El proceso de creación de la base de datos.
- Análisis y diseño del sistema
- Implementación y codificación de la aplicación.
- Fases de prueba del sistema.
- Conclusiones del proyecto y ampliaciones futuras.

2. Objetivos

La aplicación cumplirá los objetivos que se detallan a continuación:

- Automatizar al máximo el proceso logístico de la empresa.
- Agilizar el proceso de gestión y control de todo lo que comporta el mantenimiento de una empresa con un volumen de stocks reducido.
- Facilitar el registro de todos los movimientos de entrada y salida de libros a lo largo de un determinado periodo de tiempo.
- Acercar al usuario al entorno informático dotando a la aplicación de usabilidad y accesibilidad, ya que tendrá que trabajar con ella diariamente.

3. Modelo de Negocio

En este apartado, realizaremos una descripción, de la actividad que desarrolla la empresa. De esta forma, conoceremos los elementos que intervendrán, en nuestra aplicación, así como las funciones que ésta tendrá que realizar.

3.1. *Análisis del modelo*

La empresa que tenemos que informatizar es un editorial, que debido a su rápido crecimiento necesita implementar un sistema informático, para gestionar su actividad empresarial. Esta consiste en la edición de libros, que luego distribuyen entre sus clientes.

Una editorial necesita tener un control de todos los autores a los que publica, sellos editoriales con los que trabaja y el catalogo de libros editados: Tanto los que mantiene en stock, como los que tiene distribuidos entre sus diferentes clientes.

Al ser esta una empresa en expansión, necesita tener un catálogo de todos sus autores (tanto de los que ya tienen obras publicadas, como de los que están en proceso de creación), ya que son imprescindibles para el correcto funcionamiento, datos como el teléfono, número de cuenta corriente, etc.

Generalmente en una editorial, los libros se clasifican de la siguiente forma:

- Sello editorial
 - Estilo
 - Colección

Cada libro tiene una colección asociada. A su vez, una colección pertenece a un estilo. Por último un estilo, forma parte de un sello editorial. De esta forma una editorial, tiene la posibilidad de tener varios sellos, como si gestionara libros de diferentes empresas. A continuación, se muestra un ejemplo de dicha ordenación:

“GRAU ED. / Grandes Clásicos / Lengua Castellana / El quijote de la mancha”

Una de las actividades más importantes, es el mantenimiento del catálogo de libros. Esto es debido, a la gran cantidad de información, que se necesita almacenar de cada libro. A parte de datos básicos como el titulo, autor, ISBN, precio, tirada, etc., también es necesario almacenar datos técnicos del libro, por ejemplo: Tamaño de la portada, tipo de papel, tipo de letra, número de páginas en color, etc.

También se necesita tener un control, de todos los libros que tiene en almacén. Controlando de forma especial los libros que tiene un stock crítico (aquellos libros con pocos ejemplares en stock). Dentro del almacén, se registran dos tipos de movimientos:

- *Entrada de libros*, que proviene de la “imprenta”,
- *Salida de libros*, como por ejemplo, los ejemplares que se ceden a librerías y comercios para su venta.

Otra característica importante son los clientes, la empresa tiene dos tipos de clientes:

- *Clientes*, son personas físicas o fiscales, a las que se les vende libros, sin necesidad de tener un control del stock. Un cliente puede ser, por ejemplo, una persona que compra solamente un libro.
- *Distribuidores*, son las empresas que distribuyen los libros. Este es el subconjunto más importante y normalmente son librerías, tiendas especializadas o intermediarios. La editorial, necesita tener un control exhaustivo de los libros que tienen los distribuidores en stock, así como de sus ventas.

En el caso de los clientes, la editorial realiza la factura de una venta, en el momento que se produce. Mientras que a los distribuidores, se les puede facturar sus ventas conjuntamente. Por ejemplo, es posible facturar todas las ventas de un mes, en una única factura.

4. Método de desarrollo

4.1. Introducción

Existe una disciplina para el desarrollo del software, esta disciplina es la Ingeniería del Software, combinando métodos para todas las fases de desarrollo de software, mejores herramientas para desarrollar estos métodos, construcción mas poderosa de bloques para la implementación del software y una filosofía predominante de coordinación, control y gestión.

Se han buscado muchas definiciones de Ingeniería del Software, por ejemplo, la propuesta por Fritz Bauer en la primera conferencia importante dedicada al tema (Pressman, 1992):

“La Ingeniería del Software es el establecimiento y uso de principios de ingeniería robustos, orientados a obtener económicamente software que sea fiable y funcione eficientemente sobre máquinas reales.”

Pero todas refuerzan la importancia de una disciplina de ingeniería para el desarrollo del software.

Esta disciplina abarca un conjunto de tres elementos clave (métodos, herramientas y procedimientos) que facilitan al gestor controlar el proceso de desarrollo del software y suministrar a los que practiquen dicha ingeniería las bases para construir software de alta calidad de una forma productiva.

Métodos: Suministran el “cómo” construir técnicamente el software. Los métodos abarcan un amplio espectro de tareas que incluyen: planificación y estimación de proyectos, análisis de requisitos del sistema y del software, diseño de estructuras de datos y arquitectura de datos, codificación, prueba y mantenimiento.

Herramientas: Suministran un soporte automático o semiautomático para los métodos. Existen herramientas para soportar cada uno de los métodos mencionados anteriormente. Un ejemplo es la herramienta CASE que combina el software, hardware y bases de datos de la ingeniería de software para crear un entorno de ingeniería de software análogo al diseño / ingeniería asistida por computadora, para el hardware.

Procedimientos: Unen los métodos y las herramientas y facilitan un desarrollo racional y oportuno del software de computadora. Los procedimientos definen la secuencia en la que se aplican los métodos, las entregas que se requieren, los controles que ayudan a asegurar la calidad y coordinar los cambios, y las guías que facilitan a los gestores del software establecer su desarrollo.

La Ingeniería del Software, está compuesta de pasos que abarcan los métodos, herramientas y procedimientos. Estos pasos se denominan paradigmas de la Ingeniería del Software. Un paradigma para la Ingeniería del Software se elige basándose en la

naturaleza del proyecto y de la aplicación, los métodos y herramientas a usar y los controles y las entregas requeridas. En el siguiente apartado se describen con detalle cuatro paradigmas, he decidido describir estos cuatro ya que son los más importantes y los que más se usan.

El resultado final del proceso de desarrollo de un sistema de software tiene que ser un producto que tiene que ajustarse a los requisitos y se tiene que haber obtenido en los términos y con los costes previstos.

4.2. Modelos de desarrollo de software

A continuación, presentamos cuatro modelos tipo para ver, genéricamente, las posibilidades que hay. Cada modelo intenta ordenar de forma razonable las actividades de definición, desarrollo y mantenimiento de un sistema.

Los cuatro modelos que vamos a ver son: modelo secuencial, construcción de prototipos, modelo incremental y modelo en espiral.

4.2.1. Modelo secuencial o en cascada

El modelo en cascada, es el enfoque metodológico que ordena rigurosamente las etapas del ciclo de vida del software, de forma tal que el inicio de cada etapa debe esperar a la finalización de la inmediatamente anterior.

Las fases de este modelo son las siguientes:

- Ingeniería de Sistemas.
- Análisis
- Diseño del Programa.
- Codificación.
- Pruebas.
- Mantenimiento.

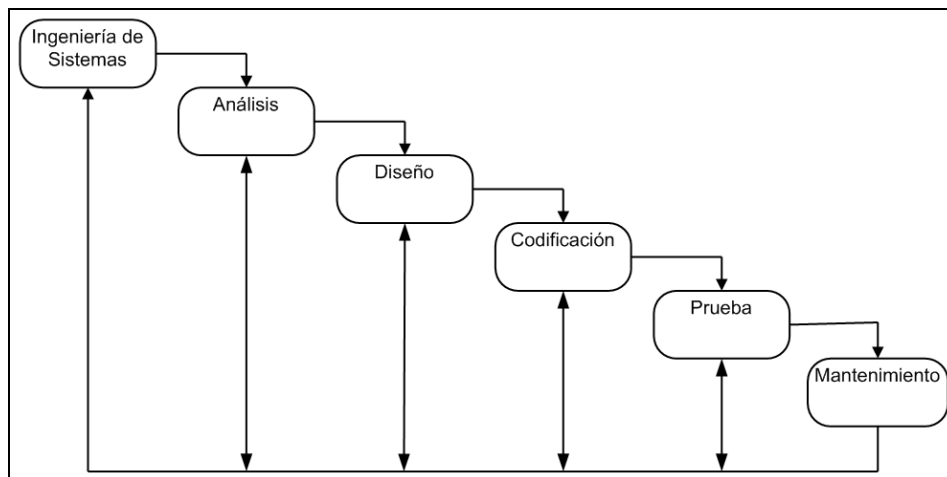


Figura 1: Modelo secuencial o en cascada

Este modelo tiene algunos problemas:

- Podemos tener ciclos de retroalimentación. Si las pruebas fallan, tal vez se tenga que retroceder y revisar fases como la ingeniería de sistemas, análisis, etc. Para evitar eso, hay que realizar pruebas unitarias.
- Desde la Ingeniería de Sistemas hasta la prueba, pasa mucho tiempo, debido a que el cliente no sabe lo que quiere realmente.

4.2.2. Modelo de prototipos o modelo de desarrollo evolutivo

Este modelo plantea soluciones al modelo anterior (Modelo secuencial). El cliente puede hacerse pronto con una idea general del producto e ir modificando aquellos aspectos que crea conveniente. El prototipo es la herramienta perfecta de comunicación entre el diseñador de aplicaciones y el cliente. Los prototipos se pueden confeccionar en lápiz y papel, o bien utilizando lenguajes de programación rápida, solamente tenemos que tener claros los requisitos.

La siguiente figura muestra el modelo de construcción de prototipos:

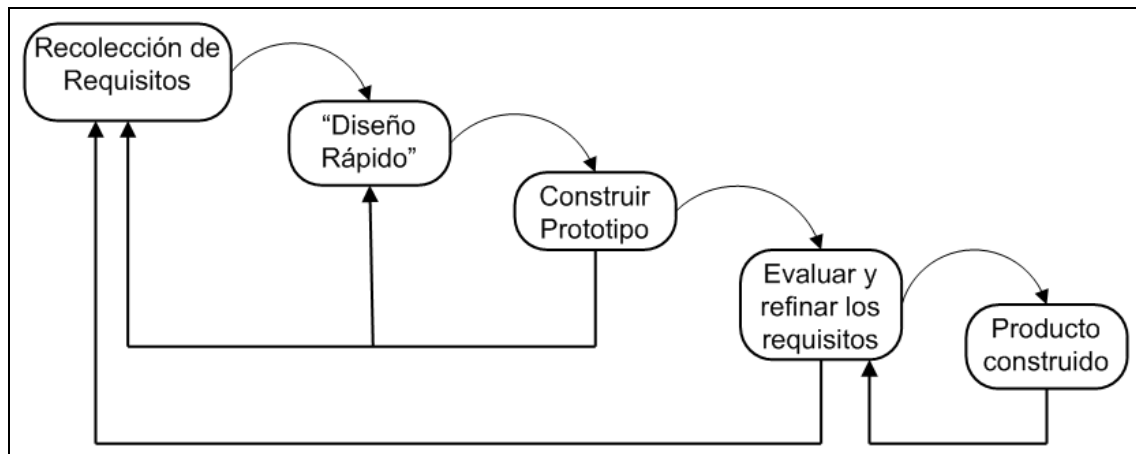


Figura 2: Construcción de prototipos

Algunos problemas de la creación de prototipos son:

- El cliente ve en funcionamiento lo que parece una versión del software, pero en realidad sólo es un prototipo y en muchos casos, debe ser reconstruido
- La implementación de estos prototipos suele ser bastante mala y se corre el riesgo, que con el paso del tiempo el técnico se familiarice con esta forma de trabajar.

4.2.3. Modelo Incremental

El sistema no se desarrolla todo a la vez, si no que se divide en partes, basándose en la funcionalidad. En cada iteración, se hace todo el proceso para desarrollar una parte concreta. Se empieza por las partes que tienen un mayor riesgo y a las que el cliente da más importancia. De esta forma se reduce el riesgo global y las funcionalidades mas importantes están probadas.

Con Este modelo se intento mantener la estructura en cascada y eliminar los prototipos.

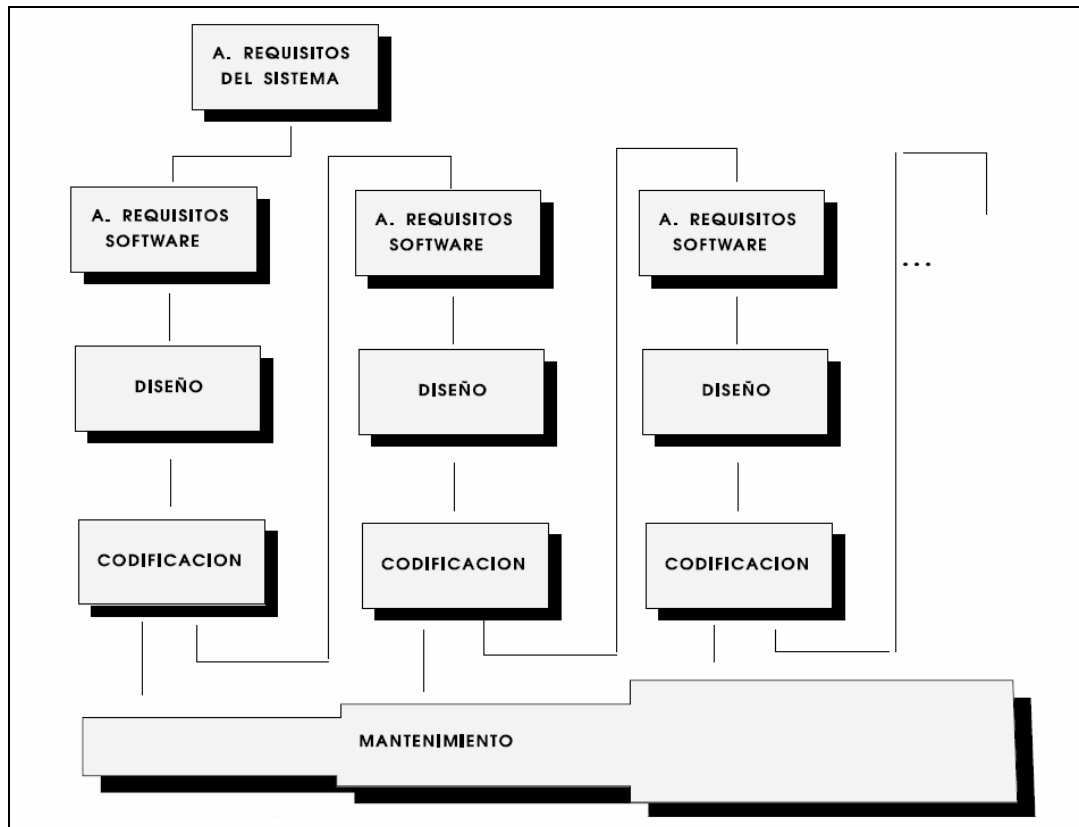


Figura 3: Modelo incremental

Fuente: www.itpl.edu.mx/publica/tutoriales/analisis

4.2.4. Modelo en espiral

El Modelo en espiral fue propuesto por Boehm (Boehm, 1988). Su idea básica es el que el desarrollo de software es un proceso iterativo.

Este modelo fue desarrollado para cubrir las mejores características del modelo secuencial como de la construcción de prototipos, añadiendo un nuevo elemento, el análisis de riesgo

En este modelo se construyen sucesivas versiones del software con la idea de complementarlas cada vez más y de manera que puedan ser evaluadas por el cliente.

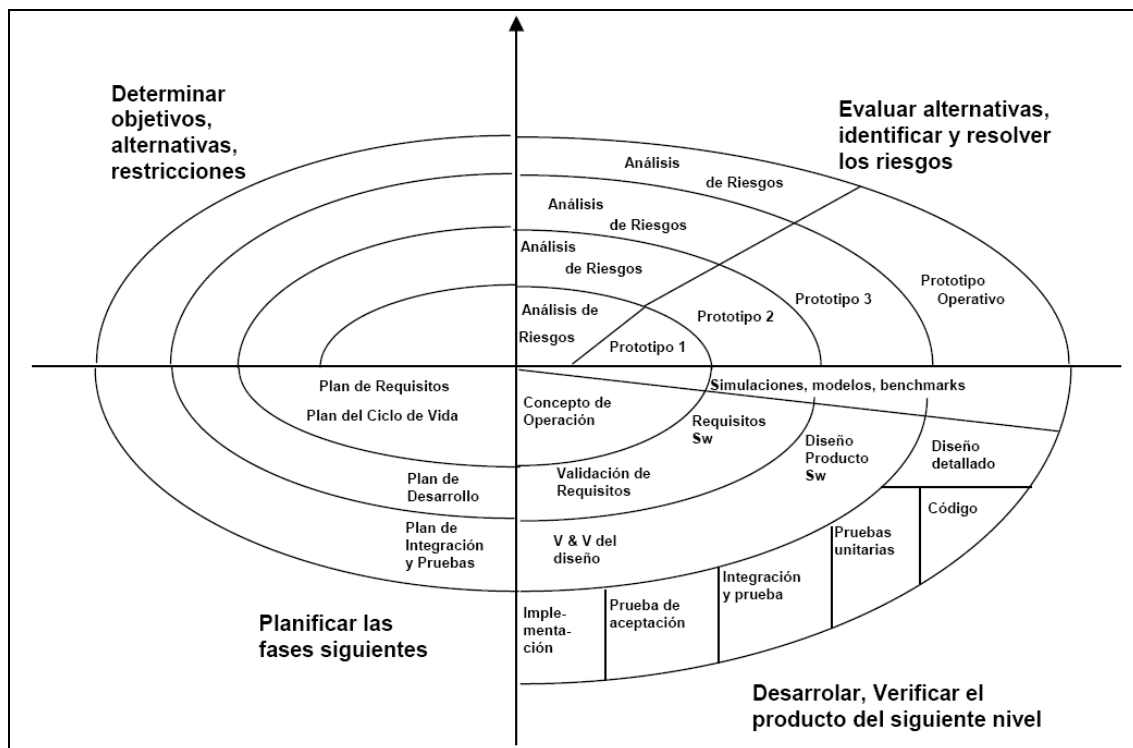


Figura 4: Modelo en espiral

Fuente: www.itpl.edu.mx/publica/tutoriales/analisis

4.2.5. Conclusiones

Tras ver los cuatro paradigmas voy a explicar por cual de ellos me decanto para la realización de esta aplicación.

En realidad, no he escogido solo uno de estos modelos, sino que he utilizado una mezcla del modelo secuencial, del modelo incremental y de la construcción de prototipos

Dado que los requisitos funcionales de la aplicación estuvieron definidos desde el inicio, he aplicado el modelo secuencial en la estructura general del proyecto.

En cambio en lo que respecta a la interfaz y a la interacción de la aplicación con el usuario, he aplicado el modelo incremental y la construcción de prototipos. Ya que la utilización de estos permite mostrar los resultados a los usuarios y una vez nos han dado su opinión, podemos modificarlos hasta lograr lo que el cliente espera.

4.3. Proceso de desarrollo del software

Sea cual sea el paradigma escogido, el proceso de desarrollo de software contiene tres fases genéricas: definición, desarrollo y mantenimiento.

Fase de definición: Esta fase se centra en el qué. Se intenta identificar qué información tiene que ser procesada, qué función y qué rendimiento se desea, cuál será la interfaz, qué restricciones de diseño existen y qué criterios de validación se necesitarán para definir un sistema correcto. De esta manera se identifican los requisitos claves del sistema y del software.

En esta fase se definen una serie de tareas:

- análisis de requisitos: Se definen los objetivos del sistema informático, las cualidades deseadas y la arquitectura del sistema informático que se necesita para el desarrollo y la explotación. Es la especificación del sistema.
- Planificación del proyecto de software: En esta etapa se analizan los riesgos, se asignan los recursos, se estiman los costes, se definen las tareas y se planifica el trabajo.
- análisis del sistema: Se representa detalladamente el sistema informático que se quiere crear. Engloba tres partes: análisis conceptual de los datos; diseño lógico de los tratamientos, que muestra los procedimientos para obtener las salidas a partir de las entradas y los datos almacenados; diseño externo de la interfaz con el usuario.

Fase de desarrollo: Esta fase se centra en el cómo. Intenta identificar cómo se tiene que diseñar las estructuras de datos y la arquitectura de software, cómo se tiene que traducir el diseño a un lenguaje de programación y cómo se tiene que realizar las pruebas.

Las tareas que se realizan en esta fase son:

- Diseño del software: se traducen los requisitos del software a un conjunto de representaciones que describen la estructura de los datos, la arquitectura, el proceso algorítmico y las características de la interfaz.
- Implementación: Las representaciones del diseño se tienen que traducir a un lenguaje de implementación.
- Prueba o validación: Una vez realizada la implementación y realizado el ejecutable para una máquina, se tiene que probar para localizar los posibles errores o defectos y verificar que funcionan de acuerdo con los requisitos especificados.

Fase de mantenimiento: Esta fase se centra en los cambios, los cuales van asociados a la corrección de los errores, a las adaptaciones o ampliaciones necesarias y a las modificaciones derivadas de los cambios que el cliente haya podido hacer en los requisitos.

Los cambios que se pueden llevar a cabo en esta fase son:

- Corrección: se puede dar el caso de que el cliente se de cuenta de defectos en el software todo y llevando a cabo una serie de actividades de garantía de calidad.
- Adaptación: Con el paso del tiempo puede ser que el entorno original con el que se desarrollo el software cambie de aquí la necesidad a la adaptación a un nuevo entorno.
- Mejora: a medida que el cliente utilice el software, le pueden surgir nuevas necesidades funcionales o de rendimiento y decida añadirlas para mejorar el software. Es por esto que nunca podemos asegurar que un programa esté acabado definitivamente y siempre es necesario un mantenimiento del mismo.

5. Análisis de requisitos y especificación del sistema

La recopilación y documentación de los requisitos es una de las tareas más importantes del desarrollo de la aplicación que vamos a construir.

Esta primera fase de construcción del proyecto es de gran relevancia, ya que una buena definición de los requisitos es muy importante para la correcta construcción de la aplicación.

5.1. Requisitos funcionales

Los requisitos funcionales describen el comportamiento deseado del software. Cada requisito funcional expresa una relación entre las entradas y salidas de la futura aplicación. También han de especificar, como se tiene que comportar esta, ante situaciones anormales.

Nuestros requisitos funcionales son los siguientes:

- Creación, modificación y eliminación de autores.
- Creación, modificación y eliminación de sellos, estilos y colecciones.
- Creación, modificación y eliminación de libros.
- Creación, modificación y eliminación de clientes.
- Creación, modificación y eliminación de distribuidores.
- Control del stock de libros del almacén.
 - Control de aquellos libros tengan un stock crítico.
- Control del stock de cada distribuidor.
- Control de los movimientos de libros entre distribuidores.
- Control de las ventas de cada distribuidor.
- Facturación de las ventas de libros.
 - Facturación a Clientes.
 - Facturación a Distribuidores.
- Realizar copias de seguridad.

5.1.1. Modelo de Casos de Uso

Un Caso de Uso se representa como un dialogo entre los usuarios y el sistema, para contemplar un proceso (que permita conseguir algunos de los objetivos de los usuarios).

Los Casos de Uso no son parte del diseño (cómo), sino parte del análisis (qué). De forma que al ser parte del análisis nos ayudan a describir qué es lo que el sistema debe hacer. Los Casos de Uso son lo qué hace el sistema desde el punto de vista del usuario. Es decir, describen un uso del sistema y cómo este interactúa con el usuario.

En esta apartado analizaremos los siguientes casos de uso, hemos escogido estos, por ser los más representativos, de la aplicación:

- Autenticación
- Alta autor
- Modificar del datos de un autor
- Eliminar un autor
- Alta Sello/Estilo/Colección
- Modificar Sello/Estilo/Colección
- Eliminar Sello/Estilo/Colección
- Alta Libro
- Modificar los datos del libro
- Eliminar un libro
- Movimiento de libros
- Venta de libros
- Facturar a un Distribuidor

Autenticación

Para acceder a nuestra aplicación el usuario tiene que introducir un nombre de usuario y una contraseña.

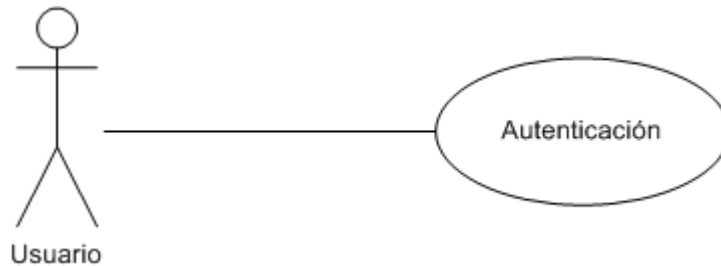


Figura 5: Caso de Uso "Autenticación"

Pre:

Post: Se ha autenticado el usuario de la aplicación.

Usuario

2. Introduce el login.
3. Introduce la contraseña.
4. Confirma los datos introducidos

Sistema

1. Muestra un formulario para insertar el login y la contraseña.
5. Comprueba si existe el usuario en la base de datos.
6. Autentica al usuario como usuario del sistema.

Tabla 1: Flujo de acontecimientos "Autenticación"

Excepciones:

4. No se ha llenado toda la información necesaria. El sistema muestra el mensaje de error correspondiente.
5. No existe ningún usuario en el sistema, con el login y la contraseña introducidos. El sistema muestra un mensaje de error y cierra la aplicación.

Alta autor

En nuestra aplicación hemos de definir los autores. Cada libro debe tener un autor. Un autor puede tener varios libros. Los atributos que utilizaremos para definir un Autor son nombre, apellidos, DNI, dirección, código postal, localidad, provincia, país, teléfono, teléfono móvil, fax, E-mail y número de cuenta corriente.

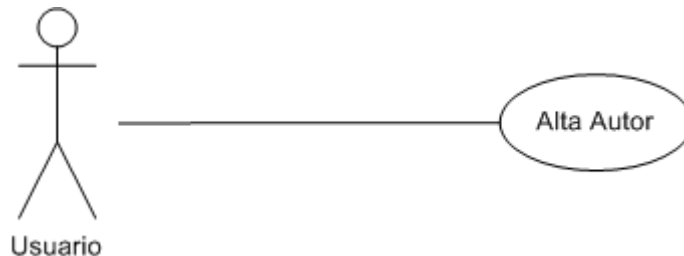


Figura 6: Caso de Uso “Alta Autor”

Pre: El usuario quiere crear un nuevo autor.

Post: Se ha creado un nuevo autor, con todos los datos necesarios.

Usuario

1. El usuario quiere crear un nuevo autor.
3. El usuario introduce los datos del autor: nombre, apellidos, DNI, dirección, código postal, localidad, provincia, país, teléfono, teléfono móvil, fax, E-mail y número de cuenta corriente
4. El usuario confirma los datos del nuevo autor.

Sistema

2. Muestra un formulario para insertar los datos de entrada.
5. El sistema almacena los datos del autor en la base de datos.

Tabla 2: Flujo de acontecimientos “Alta Autor”

Excepciones:

4. El nombre y/o los apellidos del autor son nulos.

Modificar los datos de un autor

Aquí modificamos los datos de un autor.

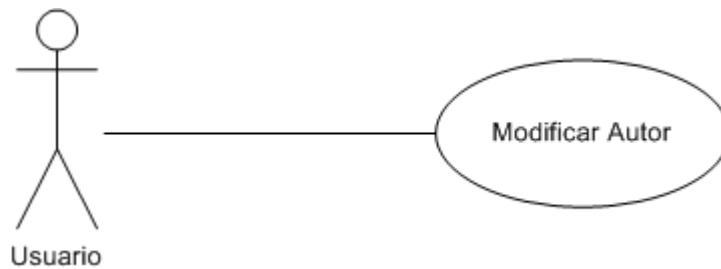


Figura 7: Caso de Uso "Modificar Autor"

Pre: El usuario quiere modificar los datos de un autor ya existente.

Post: Se han modificado los datos del autor.

Usuario

1. El usuario quiere modificar un autor.

3. El usuario selecciona el autor que quiere modificar

5. El usuario modifica los atributos del autor.

6. El usuario confirma los datos del autor.

Sistema

2. Muestra una lista con todos los autores de la aplicación.

4. El sistema carga los datos del autor, en un formulario

7. El sistema almacena los datos del autor en la base de datos.

Tabla 3: Flujo de acontecimientos "Modificar Autor"

Excepciones:

6. El nombre y/o los apellidos del autor son nulos.

Eliminar un autor

Aquí eliminamos un autor del sistema.

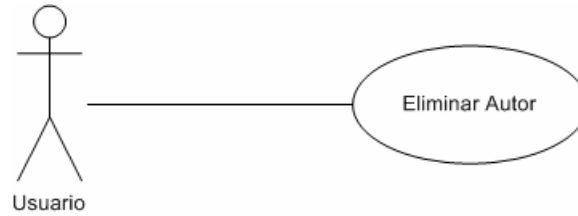


Figura 8: Caso de Uso “Eliminar Autor”

Pre: El usuario eliminar un autor existente.

Post: Se ha eliminado el autor.

Usuario

1. El usuario quiere eliminar un autor.
3. El usuario selecciona el autor que quiere eliminar
4. El usuario confirma los datos del autor.

Sistema

2. Muestra una lista con todos los autores de la aplicación.
5. El sistema borra el autor de la base de datos.

Tabla 4: Flujo de acontecimientos “Eliminar Autor”

Excepciones:

4. El autor tiene libros asociados y por lo tanto no se puede eliminar.

Alta Sello/Estilo/Colección

Los sellos, estilos y colecciones, es la forma que tiene el sistema de clasificar los libros. En este caso explicamos el alta de un sello por que todas se realizan de la misma forma.

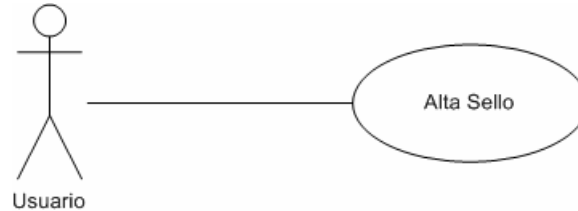


Figura 9: Caso de Uso “Alta Sello”

Pre: El usuario quiere crear un nuevo sello.

Post: Se ha creado un nuevo sello.

Usuario

1. El usuario quiere crear un nuevo sello.
3. El usuario introduce el nombre del sello.
4. El usuario confirma los datos del nuevo sello.

Sistema

2. El sistema muestra pantalla, para introducir datos.
5. El sistema almacena el nuevo sello en la base de datos.

Tabla 5: Flujo de acontecimientos “Alta Sello”

Excepciones:

4. El nombre del sello ya existe.

Modificar Sello/Estilo/Colección

En este caso explicamos como se modifica un sello, por que los estilos y colecciones, se realizan de la misma forma.

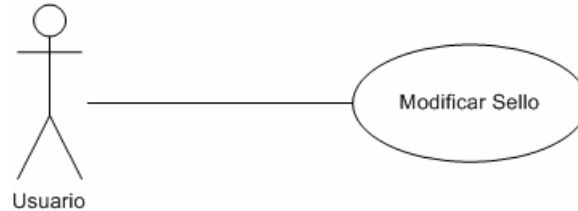


Figura 10: Caso de Uso "Modificar Sello"

Pre: El usuario quiere modificar un sello existente.

Post: Se han cambiado los datos del sello.

Usuario

1. El usuario quiere modificar un sello.
3. El usuario selecciona el sello a modificar.
5. El usuario introduce los nuevos datos del sello.
6. El usuario confirma los datos del sello.

Sistema

2. El sistema muestra por pantalla, la lista de sellos.
4. muestra una pantalla con los datos del sello.
7. El sistema almacena los datos sello en la base de datos.

Tabla 6: Flujo de acontecimientos "Modificar Sello"

Excepciones:

6. Ya existe otro sello con el nombre introducido.

Eliminar Sello/Estilo/Colección

Este Caso de Uso explica como se elimina un sello. La eliminación de un estilo o una colección, se realizan de idéntica forma.

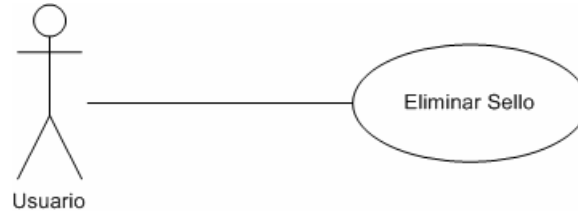


Figura 11: Caso de Uso "Eliminar Sello"

Pre: El usuario quiere eliminar un sello existente.

Post: Se ha eliminado el sello.

Usuario

1. El usuario quiere eliminar un sello.
3. El usuario selecciona un sello y clic en la opción de eliminarlo.
4. El usuario confirma la eliminación del sello.

Sistema

2. El sistema muestra pantalla con todos los sellos del sistema
5. El sistema elimina el sello de la base de datos.

Tabla 7: Flujo de acontecimientos "Eliminar Sello"

Excepciones:

4. El sello tiene libros asociados y por lo tanto no se puede eliminar.

Alta Libro

La gestión de libros, es una de operaciones más importantes, dentro de nuestro sistema. En este Caso de Uso, se explica como se crea un nuevo libro.

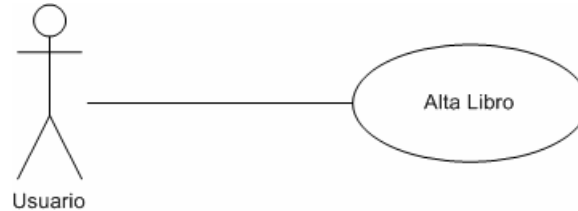


Figura 12: Caso de Uso "Alta Libro"

Pre: El sistema tiene los datos del autor del libro y el sello, estilo y colección al que pertenecerá dicho libro.

Post: Se ha creado un nuevo libro, con todos los datos necesarios.

Usuario

1. Clicka en el botón de crear un nuevo libro.

3. Introduce los datos libro: título, ISBN, tirada, precio, peso...

4. Selecciona un autor del sistema.

5. Selecciona una colección del sistema.

6. El usuario confirma los datos del nuevo libro.

Sistema

2. Muestra un formulario para insertar los datos de entrada.

7. El sistema almacena los datos del libro en la base de datos.

Tabla 8: Flujo de acontecimientos "Alta Libro"

Excepciones:

6. Alguno de los siguientes atributos del libro son nulos: Título, ISBN, Autor y Colección. El sistema mostrará un mensaje de error y el usuario tendrá que introducir valores o seleccionar algún elemento, dependiendo del caso.

7. Ya existe un libro en la base de datos con el mismo ISBN que el nuevo libro. El sistema mostrará un mensaje de error y el libro no se almacenará.

Modificar los datos del libro

En este Caso de Uso, se explica como se modifican los datos de un libro.

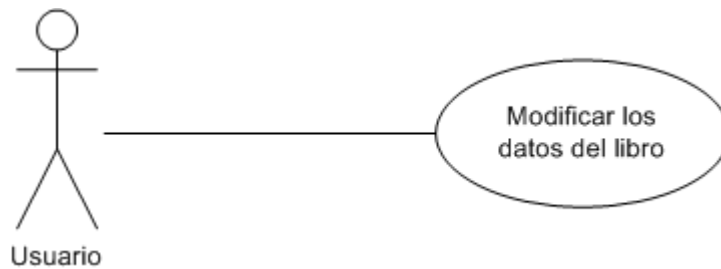


Figura 13: Caso de Uso “Modificar los datos del libro”

Pre: El sistema tiene los datos del autor del libro y el sello, estilo y colección al que pertenecerá dicho libro.

Post: Se han modificado los datos del libro.

Usuario

2. Selecciona el libro y pulsa el botón de modificar.
4. Realiza los cambios deseados
5. El usuario confirma los datos del libro.

Sistema

1. Muestra un listado con todos los libros de la aplicación.
3. Muestra una pantalla con los datos del libro.
6. El sistema almacena los datos del libro en la base de datos.

Tabla 9: Flujo de acontecimientos “Modificar los datos del libro”

Excepciones:

5. Alguno de los siguientes atributos del libro son nulos: Título, ISBN, Autor y Colección. El sistema mostrará un mensaje de error y el usuario tendrá que introducir valores o seleccionar algún elemento, dependiendo del caso.
6. Ya existe un libro en la base de datos con el mismo ISBN que el nuevo libro. El sistema mostrará un mensaje de error y el libro no se almacenará.

Eliminar un libro

A continuación se explica como eliminar un libro de la aplicación.

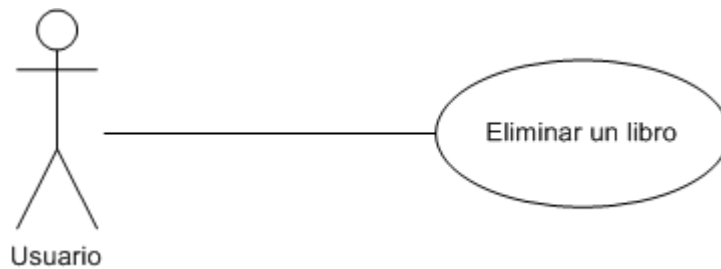


Figura 14: Caso de Uso “Eliminar un libro”

Pre:

Post: Se ha Eliminado el libro.

Usuario

2. Selecciona el libro y pulsa el botón de eliminar.
4. El usuario confirma la eliminación del libro
- .

Sistema

1. Muestra un listado con todos los libros de la aplicación.
3. Muestra una pantalla de confirmación
6. El sistema elimina el libro en la base de datos.

Tabla 10: Flujo de acontecimientos “Eliminar un libro”

Excepciones:

6. El libro no se puede eliminar por que esta siendo utilizado por el sistema. Se mostrará un mensaje de error y el no se eliminará dicho libro.

Movimiento de libros

El movimiento de libros, es una operación básica en nuestra aplicación. Gracias a ella, controlamos tanto el stock del el almacén, como el stock de cada distribuidor.

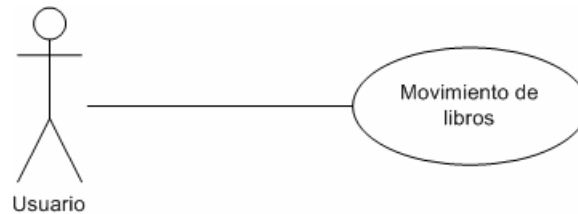


Figura 15: Caso de Uso "Movimiento de libros"

Pre: El distribuidor origen, tiene una cantidad X, de libros.

Post: El distribuidor destino, recibe parte del stock del distribuidor origen. A la vez, el distribuidor origen, actualiza su stock.

Usuario

2. Selecciona un distribuidor "origen".
4. Selecciona un libro.
5. Introduce la cantidad que desea mover.
6. Selecciona el distribuidor "destino".
7. Confirma el movimiento.

Sistema

1. Muestra un listado con todos los distribuidores de la aplicación.
3. Muestra el stock del distribuidor seleccionado.
8. Guarda el movimiento la base de datos.
9. Actualiza el stock de los distribuidores.

Tabla 11: Flujo de acontecimientos "Movimiento de libros"

Excepciones:

7. La cantidad introducida, es superior al número de ejemplares en stock, del libro seleccionado. Se mostrará un mensaje de error y el usuario tendrá que modificar la cantidad.

Venta de libros

Una vez realizado un movimiento de libros, entre distribuidores. El distribuidor destino, vende parte de su stock.

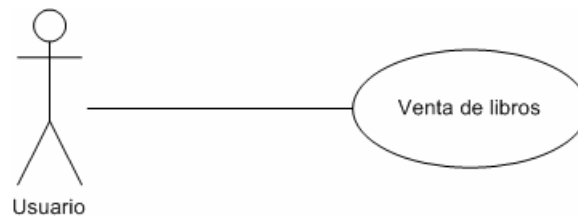


Figura 16: Caso de Uso “Venta de libros”

Pre: El distribuidor, tiene una cantidad de ejemplares de un libro.

Post: El distribuidor vende parte de su stock. Y por lo tanto, este ya puede ser facturado.

Usuario

2. Selecciona un distribuidor
4. Selecciona un libro.
5. Introduce la cantidad que desea establecer como venta.
6. Confirma el movimiento.

Sistema

1. Muestra un listado con todos los distribuidores de la aplicación.
3. Muestra el stock del distribuidor seleccionado.
7. Guarda la venta en la base de datos.
8. Actualiza el stock del distribuidor.

Tabla 12: Flujo de acontecimientos “Venta de libros”

Excepciones:

6. La cantidad introducida, es superior al número de ejemplares en stock, del libro seleccionado. Se mostrará un mensaje de error y el usuario tendrá que modificar la cantidad.

Facturar a un Distribuidor

La facturación a distribuidores es un proceso clave, dentro de la aplicación. En este caso de uso se describe:

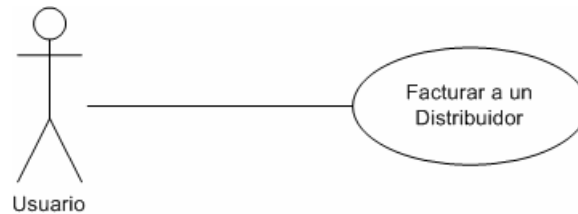


Figura 17: Caso de Uso “Facturar a un Distribuidor”

Pre: El distribuidor tiene varias ventas pendientes de facturar

Post: Se ha creado la factura y se eliminan las ventas pendientes de facturar.

Usuario

2. Selecciona el distribuidor.
4. Establece la fecha de facturación, de vencimiento y la forma de pago.
5. Selecciona las ventas que desea facturar.
6. Pulsa el botón de generar factura
8. Acepta la factura.

Sistema

1. Muestra una lista de distribuidores
3. Autocompleta los datos del distribuidor.
7. Muestra una pantalla con la vista previa de la factura.
9. Almacena los datos de la factura en la Base de datos.
10. Establece las ventas de la factura, como facturadas.

Excepciones:

6. La fecha de facturación, es anterior a la de hoy. El sistema mostrará un mensaje de error y volverá a la pantalla anterior, para que el usuario modifique la fecha de facturación.
6. La fecha de vencimiento, es anterior a la fecha de facturación. El sistema mostrará un mensaje de error y volverá a la pantalla anterior, para que el usuario modifique la fecha de vencimiento.
6. El distribuidor no ha seleccionado ventas para facturar. El sistema mostrará un mensaje de error y volverá a la pantalla anterior, para que el usuario seleccione alguna venta.

5.2. Requisitos no funcionales.

Esta aplicación, va dirigida a pequeñas empresas. Por ese motivo, en esta versión no será multipuesto. O sea, el programa estará alojado en un único ordenador, y este a su vez dispondrá de la Base de datos, en exclusividad.

Nuestro sistema informático, deberá funcionar bajo el sistema operativo Windows XP o 2000.

Además esta aplicación, debe cumplir los siguientes requisitos funcionales:

- **Ergonomía:** la aplicación se tiene que poder instalar y utilizar con facilidad, incorporando también un manual de ayuda.
- **Consistencia:** para que la aplicación sea consistente, todos los mecanismos se tendrán que utilizar siempre del mismo modo, para no confundir al usuario. Es por esto que las pantallas, estarán diseñadas de forma parecida.
- **Seguridad:** la aplicación tendrá que garantizar que no se infiltraran usuarios no permitidos. Es por este motivo, que el programa dispondrá de un control de acceso. Existirá un usuario, con el privilegio de crear nuevos usuarios.
- **Fiabilidad:** para que la aplicación sea fiable, ésta tendrá que garantizar que la entrada de datos sea en todo momento correcta, detectando posibles errores del usuario.
- **Integridad:** se podrán realizar copias de seguridad de los datos. La aplicación, deberá realizar una copia de seguridad, cada vez que se cierre. También, se pondrá a realizar cuando el usuario lo crea conveniente.
- **Solidez:** la aplicación controlará posibles errores que se producen en tiempo de ejecución y éstos serán notificados al usuario.

5.3. Especificaciones del sistema

Definiremos los requisitos máximos y mínimos para un correcto desarrollo y funcionamiento de la aplicación

5.3.1. Hardware

Los requisitos del hardware, que serán utilizados, para desarrollar la aplicación son los siguientes:

- Procesador Intel Pentium 1.7GHz
- 1 GB de memoria RAM
- Disco duro de 80 GB SATA
- Monitor de color 15.4" y resolución 1024x768
- Ratón y teclado estándar.
- Re-grabadora de CD.
- Impresora de tinta o láser.

Y los requisitos mínimos de hardware para un buen funcionamiento de la aplicación son:

- Procesador Intel Pentium 1 GHz
- 256 GB de memoria RAM
- Disco duro de 20 GB SATA
- Monitor de color 15.4" y resolución 1024x768
- Ratón y teclado estándar.
- Lector de CD.
- Impresora.

5.3.2. Software

El software utilizado para desarrollar la aplicación es:

- Microsoft Windows XP Profesional (Sistema operativo).
- Visual Studio 2005 Professional Edition (Entorno de programación). Es la herramienta utilizada para programar la aplicación.
- Microsoft SQL Server 2005 Express Edition (Sistema Gestor de Bases de datos). El SGBD utilizado en nuestra base de datos.
- SQL Server Management Studio Express (Administrador de la base de datos). Se ha utilizado para administrar la base de datos, de forma visual.
- Adobe Photoshop CS (Editor de imágenes). Esta aplicación ha sido muy útil para añadir anotaciones a las capturas de pantalla.
- Microsoft Word 2003 (Procesador de textos). Procesador de textos, usado para crear la documentación.
- Microsoft PowerPoint 2003 (Aplicación para crear y mostrar diapositivas).

5.4. Necesidades del usuario

En este apartado, describiremos las necesidades que tendrán, nuestros futuros usuarios.

5.4.1. Perfil del usuario

Los usuarios que interactuarán con GESTEDI, serán el personal de administración y producción de la empresa. Su nivel informático, será bastante elevado y su nivel profesional será medio-alto. El usuario será trabajador de la empresa, no existirán usuarios esporádicos.

Los usuarios tendrán a su disposición, un manual, donde se explicarán todas las funciones de la aplicación.

5.4.2. Las funcionalidades de la aplicación

Los objetivos de nuestra aplicación han estado definidos por el cliente mediante reuniones celebradas con este, de estas reuniones, se ha acordado que la aplicación realizar las siguientes funciones:

- Creación, modificación y eliminación de autores.
- Creación, modificación y eliminación de sellos, estilos y colecciones.
- Creación, modificación y eliminación de libros.
- Creación, modificación y eliminación de clientes.
- Creación, modificación y eliminación de distribuidores.
- Control del stock de libros del almacén.
 - Control de aquellos libros tengan un stock crítico.
- Control del stock de cada distribuidor.
- Control de los movimientos de libros entre distribuidores.
- Control de las ventas de cada distribuidor.
- Facturación de las ventas de libros.
 - Facturación a Clientes.
 - Facturación a Distribuidores.
- Creación de Copias de seguridad y un modulo restauración de la base de datos.

5.5. Análisis de la competencia

Un análisis de la competencia, nos ayudará en aspectos relativos al diseño y funcionamiento de nuestra aplicación. Es importante observar como un fabricante de software ha diseñado su aplicación, tanto a nivel de interfaz, como de funcionamiento.

El primer paso, fue analizar los productos que se pueden encontrar en Internet. El resultado de esta búsqueda, no fue satisfactorio. Se encontraron muchos productos, que cubrían parte de los requisitos del usuario. Pero estos, tenían el problema, de que parte de sus funcionalidades, no se ajustaban a los requisitos del cliente. Y por lo tanto, no se creyó oportuno, adquirir una aplicación y usar tan solo un 30% de esta. Además se tendrían que desarrollar las funciones, que dicha aplicación no realizara.

Aun así, hemos conseguido ver como trabaja la competencia.

5.6. Prototipos de papel

Durante el periodo de análisis de requisitos, se han realizado varios prototipos de papel, con la finalidad de crear, una interfaz de usuario, lo más usable posible. Estos prototipos, han sido una herramienta perfecta, para analizar las necesidades del usuario. Además, han servido de gran ayuda, a la hora de establecer una comunicación, Cliente-Analista.

6. Planificación y análisis del riesgo

El objetivo de la planificación, del proyecto de software, es proporcionar un marco de trabajo, que permita hacer estimaciones razonables, de recursos costosos y planificación temporal. Estas estimaciones, se hacen dentro de un marco de tiempo limitado, al principio de un proyecto de software, que tendremos que actualizar regularmente. Según (Agost et al, 2005).

6.1. Identificación y gestión del riesgo

El riesgo, según el *Software Engineering Institute* (SEI), es la posibilidad de sufrir pérdidas. El simple conocimiento de los riesgos de una actividad, ya supone una ventaja, facilitando un estado de alerta sobre los mismos. Además, hace que disminuyan sus consecuencias, en caso de producirse.

A continuación, mencionaremos los riesgos que podemos encontrar en un proyecto:

- **Riesgos de proyecto:** Identifican los problemas presupuestarios, de agenda, de personal, de recursos, del cliente y de los requisitos.
- **Riesgos técnicos:** Identifican problemas de diseño, implementación, interfaz, verificación y mantenimiento. También son factores de riesgo la ambigüedad de la especificación, incertidumbre técnica, obsolescencia técnica y tecnológica.
- **Riesgos de negocio:** Identifica los problemas que ponen en peligro la viabilidad del proyecto. Estos son:
 - **Riesgos de mercado,** representan la construcción de un producto excelente que nadie quiere.
 - **Riesgos de gestión,** la pérdida del apoyo de los gestores, debido a un cambio de enfoque o a un cambio personal.
 - **Riesgos de presupuesto,** son las pérdidas presupuestarias o de personal.

El análisis del riesgo, consiste en enumerar los riesgos concretos, que puede sufrir nuestro proyecto, de cada riesgo debe saberse, la probabilidad de que este sea real y las consecuencias, de los problemas asociados.

Ahora, identificamos estos riesgos en nuestro proyecto:

- **Riesgos del proyecto:** Es el riesgo más importante al que está expuesto el proyecto.
 - **Falta de formación en las herramientas de desarrollo:** Para reducir este riesgo, he elegido codificar la aplicación usando *Visual Basic 2005* y usar *SQL server 2005 express* para crear la base de datos. Debido a que dispongo de un buen dominio de dichas herramientas.
 - **Problemas de agenda:** Es difícil realizar la estimación del tiempo, para realizar la aplicación, lo que nos puede ocasionar problemas de agenda. Una vez hemos detectado este error, debemos modificar la planificación y comprobar que el plazo de entrega del producto, queda dentro de los límites establecidos.
 - **Problemas de requisitos:** El cambio de los requisitos por parte del cliente, implicará la modificación del diseño de la aplicación, así como los cambios pertinentes en la implementación.
- **Riesgos técnicos:** Se harán en el diseño de la interfaz.
 - **Problemas de diseño:** Pueden surgir problemas con las modificaciones de los requisitos, en las fases avanzadas de la aplicación por parte del cliente, lo que puede implicar cambios en el diseño que ya realizado. Para evitar este problema, se ha mantenido una comunicación constante con el cliente.
 - **Problemas de codificación:** Para prevenir este riesgo y solucionar rápidamente posibles errores de código, se ha comentado todo el código, usando el sistema comentarios y documentación XML, de la versión 2005 de Visual Basic.
- **Riesgos de negocio:**
 - **Riesgos de distribución del producto:** Si el producto no puede ser distribuido como inicialmente se tenía previsto, se deberán encontrar nuevos contactos o canales de distribución de productos informáticos.

6.2. Planificación

La planificación del proyecto consiste en realizar una distribución anticipada de las tareas que se van a desarrollar. Para ello se debe realizar una estimación y distribución de cada tarea determinando las prudencias entre estas.

En esta tabla se muestra las etapas del proyecto y su duración:

TAREA	DURACIÓN (horas)	Precedencia entre actividades
1. Requisitos	27,5	
1.1 Recogida de requisitos.	12,5	
1.2 Análisis de requisitos.	15	1.1
2. Planificación de las tareas	5	1
3. Base de Datos	37,5	1
3.1 Análisis y diseño de la base de datos.	37,5	
4. Diseño de la aplicación	50	1
4.1 Diseño de la estructura del programa.	25	
4.2 Diseño de la interfaz de usuario.	20	
4.3 Diseño de los informes.	5	
5. Implementación	200	4
5.1. Aprendizaje de las herramientas de desarrollo.	20	
5.2. Implementación y codificación de la aplicación.	170	
5.2.1. Creación de la pantalla principal.	20	
5.2.2. Modulo de gestión Autores.	15	
5.2.3. Modulo de gestión Sellos estilos y colecciones.	15	
5.2.4. Modulo de gestión Libros.	20	5.2.2. Y 5.2.3.
5.2.5. Modulo de gestión del Almacén.	10	5.2.4.
5.2.6. Modulo de gestión de los distribuidores.	15	
5.2.7. Modulo de gestión de los clientes.	15	
5.2.8. Modulo de movimientos de libros.	20	5.2.4. Y 5.2.6.
5.2.9. Modulo de movimientos de facturación.	25	5.2.7. Y 5.2.8.
5.2.10. Implementación de los informes.	15	
5.3. Elaboración de la ayuda.	10	
6. Evaluación y pruebas	20	5
7. Documentación	50	5
7.1. Elaboración del manual de usuario	10	
7.2. Elaboración de la documentación	40	6
COSTE TOTAL	390	

Tabla 13: Tareas del proyecto

6.3. Presupuesto de proyecto

Partiendo de los datos obtenidos en la tabla 13, vamos a realizar un presupuesto, donde se detallan todos los gastos derivados, de la elaboración del proyecto.

- **Personal**, presupuesto referente a los salarios del equipo humano, que trabaja en la aplicación. Según la tabla anterior, el número total de horas es 390. aunque estas se computan de diferente forma:

- Las horas dedicadas a las tareas 1, 2, 3, 4 y 7 (de la tabla 13), son tareas de análisis, y por lo tanto, su precio por hora se 47,01 €¹.
- Mientras que las tareas 5 y 6, son tareas de codificación y pruebas, su precio será 33,68 €².

Por lo tanto. $(170 \text{ horas} * 47,01 \text{ €}) + (220 \text{ horas} * 33,68 \text{ €}) = \mathbf{15.401,30 \text{ €}}$

- **Inventariable**, presupuesto referente a los equipos informáticos, que tenemos utilizar. Suponiendo que un equipo completo (PC, pantalla, impresora, scanner, etc.), vale 1.500€ y que se amortiza en 4 años:

- Se estima que el proyecto tendrá una duración de 5 meses (4 horas de dedicación diaria de lunes a viernes).

Por lo tanto, $(5/48) \text{ meses} * 1.500\text{€} = \mathbf{156,25 \text{ €}}$

- **Licencias del software** en este apartado, solamente tenemos que aplicar el coste de la licencia del Visual Studio 2005 professional edition, puesto que el sistema gestor de bases de datos Microsoft SQL Server 2005 Express Edition es gratuito.

- Licencia de Visual Studio 2005 professional edition = **895.00 €**

- **Fungible**, en este apartado se contemplan todos los gastos correspondientes al material fungible: papel, CD's, cartuchos de tinta...

MATERIAL	PRECIO / UNI.	PRECI TOTAL
3 paquetes de folios	4,10 €	12,30 €
1 bobina 10 CD's	6 €	6 €
1 cartucho tinta HP 846 B/N	27,74 €	27,74 €
1 cartucho tinta HP 846 COLOR	35,20 €	35,20 €
3 bolígrafos marca Pilot ball	1,75 €	5,25 €
TOTAL		86,49 €

Tabla 14: Presupuesto material fungible

1. Sueldo mínimo de un Analista-programador y Diseñador de página Web, según (BOE núm. 11, 12/1/2007).

2. Sueldo mínimo de un Programador senior, Jefe de operación y Programador en Internet, según (BOE núm. 11, 12/1/2007).

- **Viajes y dietas**, nuestro cliente esta ubicado en Huesca.
 - La distancia desde Lleida a Huesca es de 100 Km. y el precio por kilómetro es de 0,35€. Por lo tanto cada viaje le costará al cliente 70€.
 - Cada viaje realizado incluirá una dieta de 38€.

Se han realizado 4 desplazamientos, por lo tanto $4 * (70 \text{ €} + 38 \text{ €}) = 432 \text{ €}$

- **Overhead**, Considerando todos los gastos indirectos hemos llegado a un acuerdo con el cliente, para que el overhead sea el 10% sobre el total del presupuesto.

Por lo tanto, el presupuesto se establece de la siguiente forma:

CONCEPTO	PRECIO
Personal	15.401,30 €
Inventariable	156,25 €
Licencias del software	895,00 €
Fungible	86,49 €
4 Viajes y 4 dietas	432,00 €
Subtotal:	16.971,04 €
Overhead (10%):	1.697,10 €
Total del presupuesto:	18.668,14 €

Tabla 15: Presupuesto

En la tabla anterior no se ha aplicado el IVA, este será el 16%.

7. Base de datos

La base de datos es un elemento clave, dentro del sistema informático. En este punto, se analizará su estructura interna.

Como hemos comentado en el presupuesto (apartado 6.3. Presupuesto de proyecto), para la creación y el mantenimiento de la base de datos, se ha usado Microsoft SQL Server 2005 Express Edition.

7.1. Modelo Entidad-Relación

En el Proceso de diseño de una base de datos se obtiene el esquema conceptual, en el que se define todos los datos que intervienen en el problema y sus relaciones. El modelo conceptual más conocido por modelo entidad-relación, fue propuesto por Peter Chen en 1976, para la presentación gráfica de los problemas que forman parte del mundo real.

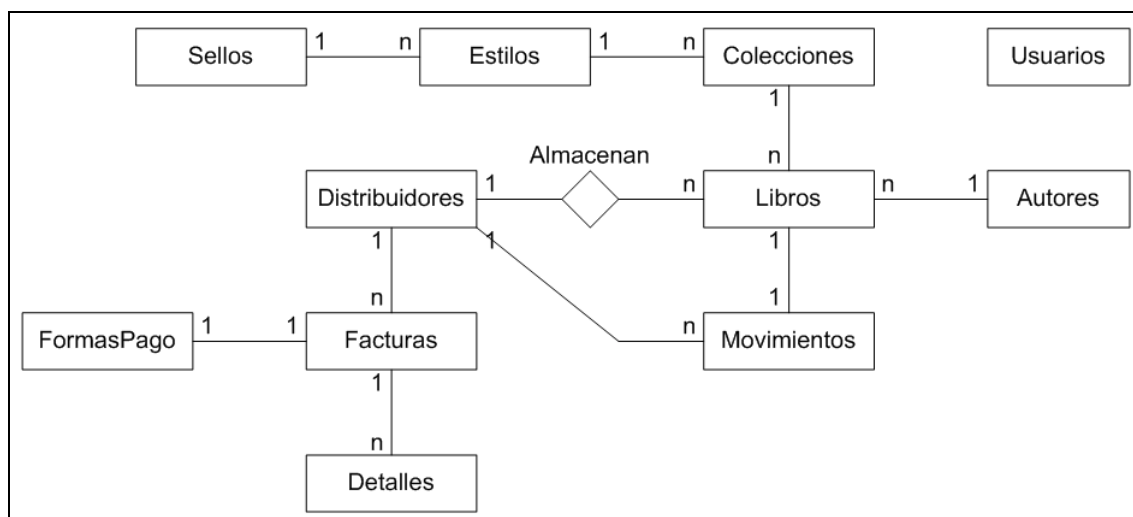


Figura 18: Modelo Entidad-Relación

7.2. Modelo Relacional

El modelo relacional es un modelo simple y potente que sirve para representar problemas. La sencillez del modelo ha facilitado la construcción de lenguajes de consulta e interfaces de fácil manejo. El elemento principal del modelo relacional es la relación. Una base de datos relacional está formada por un conjunto de relaciones. A continuación, se representa mediante un diagrama de bases de datos (generada por SQL Server 2005), la base de datos que utilizará el programa:

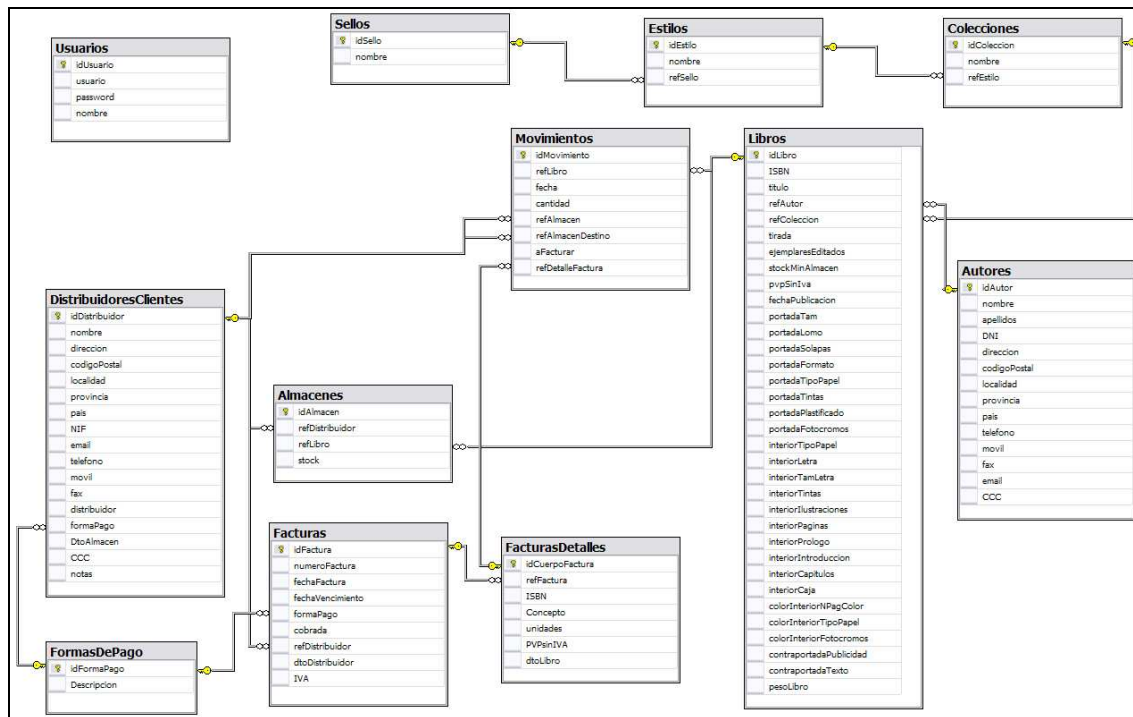


Figura 19: Diagrama de la base de datos

Comentarios acerca del diagrama:

- Se ha optado por unir en una sola tabla (*DistribuidoresClientes*) las entidades Clientes y Distribuidores, para facilitar la facturación. De esta forma la tabla Facturas sólo necesita un clave ajena a dicha tabla. Para saber si una tupla es un cliente, o un distribuidor, se ha añadido un campo booleano, llamado *distribuidor* (si su contenido es *True*, se trata de un distribuidor). Esto también, facilitara la implementación de la aplicación.
- Se ha establecido una relación entre *DistribuidoresClientes* y *FormasDePago*, para almacen la forma de pago, preferida por un distribuidor o cliente. Aun así, el usuario de la aplicación podrá elegir la forma de pago en cada factura.
- La empresa será tratada (a nivel de almacenamiento de datos) como un distribuidor más. Para consultar los libros del almacen de la empresa, el programa deberá listar todas las filas con *refDistribuidor* igual a 1.

- La tabla *Movimientos*, es una de las tablas más importantes de la base de datos. En ella se registran las siguientes operaciones:
 - Carga inicial de libros, cuando llegan libros de la imprenta a la editorial.
 - Movimiento entre distribuidores, se registra cuando un distribuidor cede un número de libros a otro. Por este motivo, existen dos claves ajenas a *DistribuidoresClientes* (*refAlmacen* y *refAlmacenDestino*).
 - Venta de libros, cuando un distribuidor vende un determinado número de libros. En este caso, en campo *aFacturar* de tipo booleano, será *True*.
- La relación *FacturasDetalles – Movimientos*, nos indicara en una venta, si esta ha sido facturada y a que línea de factura pertenece.

7.3. Diccionario de datos

Tabla *Sellos*:

Campo	Tipo de dato	Clave principal	Calve ajena
idSello	int	Sí	No
nombre	varchar(50)	No	No

Tabla 16: Tabla “Sellos” de la base de datos

Tabla *Estilos*:

Campo	Tipo de dato	Clave principal	Calve ajena
idEstilo	int	Sí	No
nombre	varchar(50)	No	No
refSello	int	No	Sí

Tabla 17: Tabla “Estilos” de la base de datos

Tabla *Colecciones*:

Campo	Tipo de dato	Clave principal	Calve ajena
idColeccion	int	Sí	No
nombre	varchar(50)	No	No
refEstilo	int	No	Sí

Tabla 18: Tabla “Colecciones” de la base de datos

Tabla *Autores*:

Campo	Tipo de dato	Clave principal	Calve ajena
idAutor	int	Sí	No
nombre	varchar(50)	No	No
apellidos	varchar(100)	No	No
DNI	nchar(10)	No	No
direccion	varchar(50)	No	No
codigoPostal	nchar(5)	No	No
localidad	varchar(50)	No	No
provincia	varchar(50)	No	No
pais	varchar(50)	No	No
telefono	nchar(10)	No	No
movil	nchar(10)	No	No
fax	nchar(10)	No	No
email	varchar(150)	No	No
CCC	varchar(20)	No	No

Tabla 19: Tabla “Autores” de la base de datos

Tabla *Libros*:

Campo	Tipo de dato	Clave principal	Clave ajena
idLibro	int	Sí	No
ISBN	varchar(13)	No	No
titulo	varchar(50)	No	No
refAutor	int	No	Sí
refColeccion	int	No	Sí
tirada	int	No	No
ejemplaresEditados	int	No	No
stockMinAlmacen	int	No	No
pvpSinIva	money	No	No
fechaPublicacion	datetime	No	No
portadaTam	varchar(50)	No	No
portadaLomo	int	No	No
portadaSolapas	varchar(50)	No	No
portadaFormato	varchar(50)	No	No
portadaTipoPapel	varchar(50)	No	No
portadaTintas	varchar(50)	No	No
portadaPlastificado	int	No	No
portadaFotocromos	int	No	No
interiorTipoPapel	varchar(50)	No	No
interiorLetra	nchar(10)	No	No
interiorTamLetra	int	No	No
interiorTintas	varchar(50)	No	No
interiorIlustraciones	varchar(50)	No	No
interiorPaginas	int	No	No
interiorPrologo	bit	No	No
interiorIntroduccion	bit	No	No
interiorCapitulos	int	No	No
interiorCaja	varchar(50)	No	No
colorInteriorNPagColor	int	No	No
colorInteriorTipoPapel	varchar(50)	No	No
colorInteriorFotocromos	int	No	No
contraportadaPublicidad	bit	No	No
contraportadaTexto	bit	No	No
pesoLibro	float	No	No

Tabla 20: Tabla “Libros” de la base de datos

Tabla *DistribuidoresClientes*:

Campo	Tipo de dato	Clave principal	Clave ajena
idDistribuidor	int	Sí	No
nombre	varchar(50)	No	No
direccion	varchar(50)	No	No
codigoPostal	nchar(5)	No	No
localidad	varchar(50)	No	No
provincia	varchar(50)	No	No
pais	varchar(50)	No	No
NIF	nchar(10)	No	No
email	varchar(50)	No	No
telefono	nchar(10)	No	No
movil	nchar(10)	No	No
fax	nchar(10)	No	No
distribuidor	bit	No	No
formaPago	int	No	No
DtoAlmacen	int	No	No
CCC	varchar(20)	No	No
notas	text	No	No

Tabla 21: Tabla “DistribuidoresClientes” de la base de datos

Tabla *Almacenes*:

Campo	Tipo de dato	Clave principal	Clave ajena
idAlmacen	int	Sí	No
refDistribuidor	int	No	Sí
refLibro	int	No	Sí
stock	int	No	No

Tabla 22: Tabla “Almacenes” de la base de datos

Tabla *FormasDePago*:

Campo	Tipo de dato	Clave principal	Clave ajena
idFormaPago	int	Sí	No
Descripcion	nvarchar(50)	No	No

Tabla 23: Tabla “FormasDePago” de la base de datos

Tabla *Usuarios*:

Campo	Tipo de dato	Clave principal	Clave ajena
idUsuario	int	Sí	No
usuario	varchar(50)	No	No
password	nchar(10)	No	No
nombre	varchar(50)	No	No

Tabla 24: Tabla “Usuarios” de la base de datos

Tabla *Movimientos*:

Campo	Tipo de dato	Clave principal	Clave ajena
idMovimiento	int	Sí	No
refLibro	int	No	Sí
fecha	datetime	No	No
cantidad	int	No	No
refAlmacen	int	No	Sí
refAlmacenDestino	int	No	Sí
aFacturar	bit	No	No
refDetalleFactura	int	No	Sí

Tabla 25: Tabla “Movimientos” de la base de datos

Tabla *Facturas*:

Campo	Tipo de dato	Clave principal	Clave ajena
idFactura	int	Sí	No
numeroFactura	nvarchar(20)	No	No
fechaFactura	datetime	No	No
fechaVencimiento	datetime	No	No
formaPago	int	No	No
cobrada	bit	No	No
refDistribuidor	int	No	Sí
dtoDistribuidor	int	No	No
IVA	int	No	No

Tabla 26: Tabla “Facturas” de la base de datos

Tabla *FacturasDetalles*:

Campo	Tipo de dato	Clave principal	Clave ajena
idCuerpoFactura	int	Sí	No
refFactura	int	No	Sí
ISBN	varchar(13)	No	No
Concepto	varchar(50)	No	No
unidades	int	No	No
PVPsinIVA	money	No	No
dtoLibro	real	No	No

Tabla 27: Tabla “FacturasDetalles” de la base de datos

8. Diseño del sistema

En esta fase el objetivo es documentar, lo tiene que hacer nuestra aplicación. Es importante definir como será la estructura de esta. Por este motivo, presentamos el análisis de tareas y la estructura de la información.

8.1. Análisis jerárquico de tareas (HTA)

El objetivo análisis jerárquico de tareas (HTA), es el de realizar una descripción las tareas en términos de operaciones y planes. Dicha descripción, se muestra en forma de árbol.

Las operaciones son actividades que debe realizar un usuario, para conseguir un objetivo.

Los planes son una descripción de las condiciones que se tiene dar, cuando se realiza cada una de las actividades. Según (Guasch et al, 2006).

A continuación, se representaran las tareas más importantes de la aplicación:

- Dar de alta un nuevo libro.
- Modificar los datos de un libro.
- Eliminar un libro.
- Cargar ejemplares de un libro.
- Movimiento de libros entre distribuidores.

Tarea 1: Dar de alta un nuevo libro.

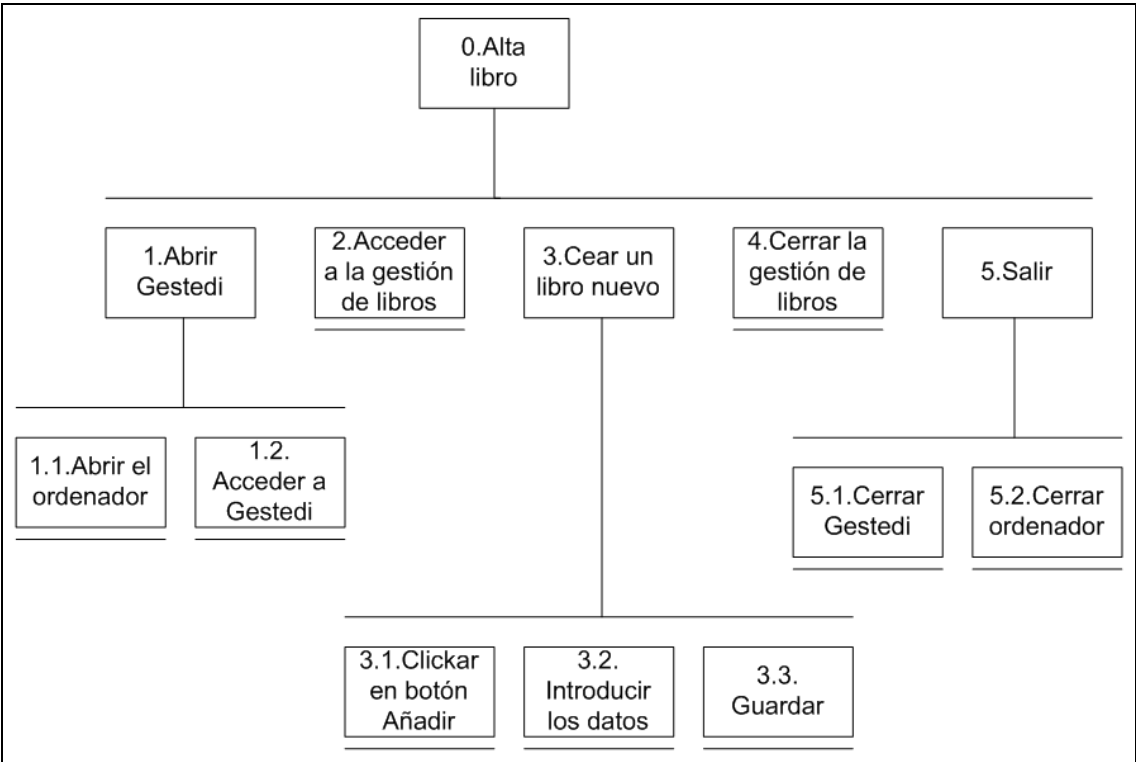


Figura 20: Tarea “Alta Libro”

Plan 0: Hacer 1, 2, 3, 4 y 5.

Plan 1: Hacer 1.1 y 1.2.

Plan 3: Hacer 3.1, 3.2 y 3.3.

Plan 5: Hacer 5.1 y 5.2.

Tarea 2: Modificar los datos de un libro.

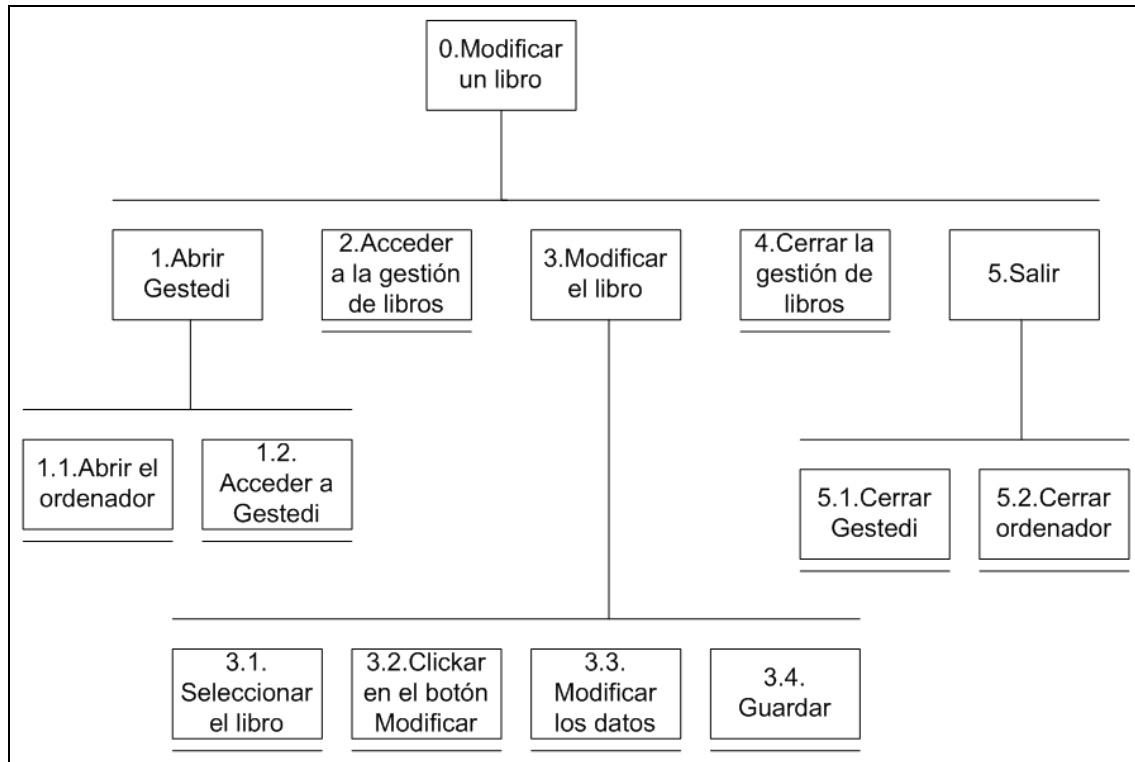


Figura 21: Tarea "Modificar Libro"

Plan 0: Hacer 1, 2, 3, 4 y 5.

Plan 1: Hacer 1.1 y 1.2.

Plan 3: Hacer 3.1, 3.2, 3.3 y 3.4.

Plan 5: Hacer 5.1 y 5.2.

Tarea 3: Eliminar un libro.

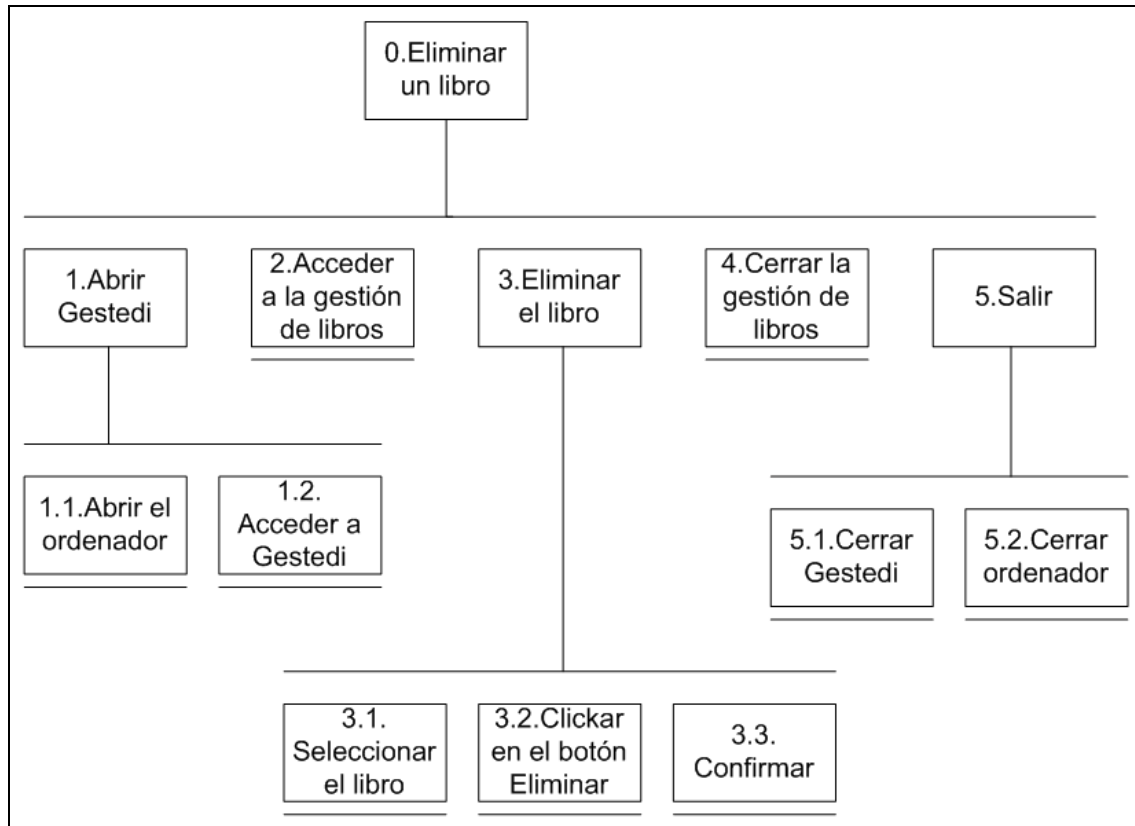


Figura 22: Tarea “Eliminar un libro”

Plan 0: Hacer 1, 2, 3, 4 y 5.

Plan 1: Hacer 1.1 y 1.2.

Plan 3: Hacer 3.1, 3.2 y 3.3.

Plan 5: Hacer 5.1 y 5.2.

Tarea 4: Cargar ejemplares de un libro.

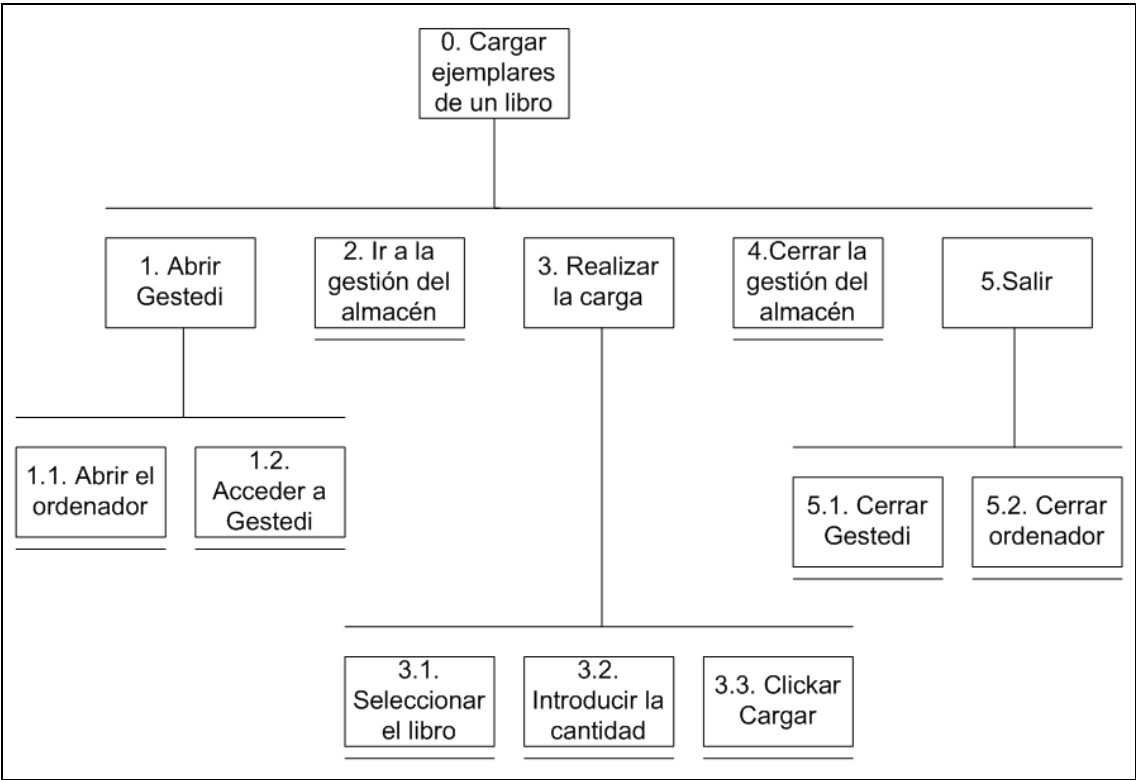


Figura 23: Tarea “Cargar Ejemplares”

Plan 0: Hacer 1, 2, 3, 4 y 5.

Plan 1: Hacer 1.1 y 1.2.

Plan 3: Hacer 3.1, 3.2 y 3.3.

Plan 5: Hacer 5.1 y 5.2.

Tarea 5: Movimiento de libros entre distribuidores.

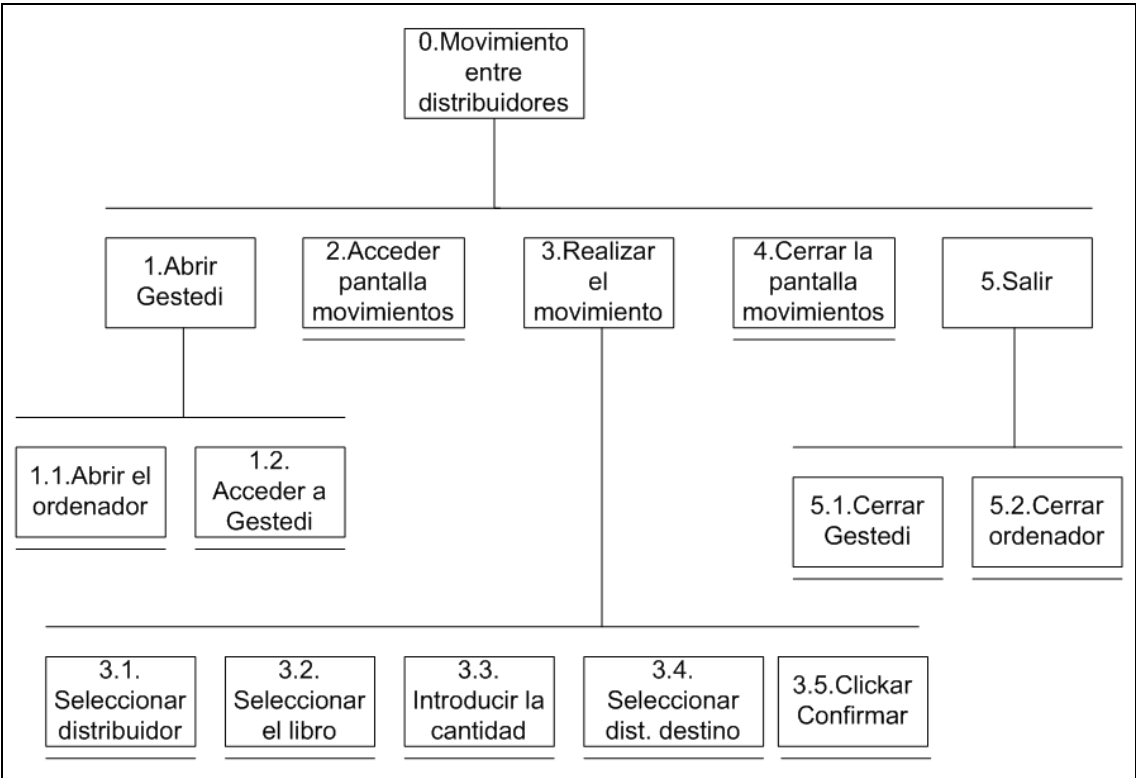


Figura 24: Tarea “Movimiento entre distribuidores”

- Plan 0:** Hacer 1, 2, 3, 4 y 5.
Plan 1: Hacer 1.1 y 1.2.
Plan 3: Hacer 3.1, 3.2, 3.3, 3.4 y 3.5.
Plan 5: Hacer 5.1 y 5.2.

8.2. Diseño externo

En la elaboración del diseño externo de la aplicación hemos intentado seguir un mismo criterio para todas las pantallas. El diseño de estas, se puede consultar en el anexo 1, manual de usuario. A continuación describiremos el modelo de la interfaz.

8.2.1. Modelo de la interfaz

La interfaz es el medio de comunicación entre el usuario y la aplicación.

Una de las características más importantes que tiene que tener una aplicación es que sea usable, es decir, que sea útil, fácil de utilizar, fácil de aprender y que el usuario se sienta bien trabajando con ella. También tiene que ser accesible, es decir, que permita tanto el acceso por ratón como por teclado. En resumen, que el diseño tiene que estar centrado en el usuario y sus necesidades.

Las funcionalidades de esta aplicación estarán divididas en cinco menús diferentes:

Menú Pantallas:

- Pantalla de Autores.
- Pantalla de Sello, estilos y colecciones.
- Pantalla de Libros.
- Pantalla de Almacén.
- Pantalla de Distribuidores.
- Pantalla de Clientes.
- Pantalla de Movimientos.
- Pantalla de Facturas.

También podemos acceder a las pantallas, por la barra de herramientas de la aplicación:



Figura 25: barra de herramientas de la aplicación

Menú Listados:

- Listado de Autores.
- Listado de Libros.
- Listado de Clientes.
- Listado de Clientes con sus compras.
- Listado de Distribuidores.
- Listado de Distribuidores con su stock.
- Listado de Almacén.
- Listado de Movimientos de libros.
- Listado de Facturas.

Menú Ver:

- Barra de herramientas.
- Barra de estado.

Menú Herramientas:

- Datos de la empresa.
- Copias de seguridad.
- Restaurar la base de datos.
- Mostrar información rápida.
- Calculadora.
- Bloc de notas.

Menú de Ventanas:

- Cascada.
- Mosaico horizontal.
- Mosaico vertical.
- Cerrar todas.

Guía de estilo de las pantallas de la aplicación:

En este apartado realizaremos una breve descripción, de los rasgos más importantes del diseño de las pantallas de la aplicación. Para extender esta información, le remitimos al *Anexo 1 Manual de usuario*.

Las pantallas de Autores, Libros, Distribuidores y Clientes, siguen un mismo diseño. Esto facilitará al usuario tanto en el proceso de aprendizaje, como a la hora de operar con ellas. Como podemos observar (en la figura 26), se distinguen varias zonas dentro de la pantalla:

1. En la parte central de la pantalla, se muestra un listado, con las columnas más importantes. Luego, realizando doble clic, podremos ver la ficha asociada a esa fila. Si fuera necesario, el listado dispone de barras de desplazamiento, para poder listar todos los datos. Otra característica importante, es que podemos

- ordenar los datos por la columna que se desee. También se podrá modificar el orden de las columnas.
2. Todas las pantallas de la aplicación tienen un botón de cerrar en la parte inferior-derecha de la pantalla.
 3. En la parte superior encontramos una barra de herramientas, con la que podremos realizar todas las operaciones necesarias para el mantenimiento del catálogo. Colocando toda la funcionalidad de la pantalla de forma ordenada, en la barra de herramientas, de esta forma evitamos la dispersión de botones por el formulario y damos mayor importancia a los datos. Esta barra de herramientas se puede mover, si preferimos otra ubicación.
 4. Las pantallas se pueden maximizar y los controles se redimensionan ocupando toda la pantalla. Esto hace que se puedan mostrar por pantalla un mayor número de información.

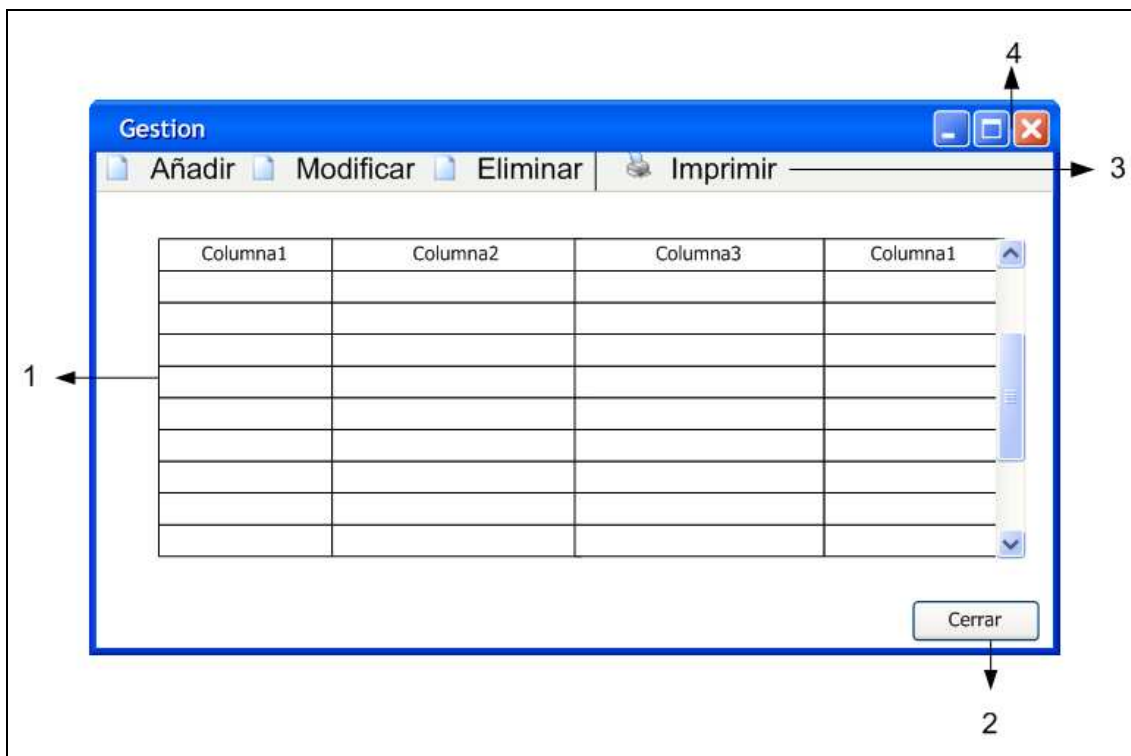


Figura 26: Diseño de las pantallas de Gestión

El resto de pantallas de la aplicación, que por su naturaleza son distintas, también siguen la misma distribución, en la medida de lo posible. Si la pantalla solamente tiene un botón o dos, se ha optado por eliminar la barrar de herramientas y colocarlos en la parte inferior-izquierda de la pantalla.

A la hora de diseñar la pantalla de Sellos, estilos y colecciones, se ha optado por “esconder” su funcionalidad. Esto es debido a que el contenido de esta pantalla, rara vez será modificado. Por ejemplo, para añadir un sello (estilo o colección), el usuario tendrá que acceder al menú contextual de la lista de sellos.

Los mensajes de información de la aplicación se mostrarán por la barra de estado de la pantalla principal.

9. Implementación

Como hemos explicado anteriormente, para la realización de nuestra aplicación, necesitaremos almacenar nuestros datos, en una Base de datos. De igual forma, necesitaremos una plataforma de desarrollo, que nos permita interactuar con la base de datos.

9.1. Herramientas de desarrollo

En este apartado describiremos las herramientas utilizadas en la implementación del código de la aplicación.

9.1.1. ¿Qué es la plataforma .NET?

La plataforma .NET es una capa de software intermedio, que se coloca entre las aplicaciones y el sistema operativo, cuya finalidad es simplificar el desarrollo del nuevo software, aportando todos los servicios necesarios para ello. De esta forma, una aplicación diseñada para la plataforma .NET puede ejecutarse virtualmente en cualquier equipo donde esté instalada la propia plataforma, sin importar el sistema operativo. Esto es posible, gracias a que la plataforma, no es una simple biblioteca de clases, si no que ofrece asimismo servicios de más bajo nivel, indispensables para los compiladores y la ejecución del código.

Es la responsable de recuperar el código de la aplicación que se desea ejecutar, verificar dicho código, compilarlo a código nativo si fuese preciso y ocuparse de aspectos como su alojamiento en un dominio de aplicación, administrar la memoria que necesite, liberar los elementos que no sean útiles, etc. Hasta cierto punto, podríamos decir que la plataforma .NET hace las veces de sistema operativo para nuestras aplicaciones.

Además, la plataforma .NET pone a nuestro alcance posibilidades no contempladas en el sistema operativo, como es el desarrollo de aplicaciones Web (ASP.NET) o la conexión con bases de datos (ADO.NET).

9.1.2. Visual Basic 2005

Visual Basic no es un lenguaje nuevo, al menos en cuanto a su origen, pero si en cuanto a su evolución. Con la llegada de la plataforma .NET, Microsoft adaptó el lenguaje Visual Basic a dicha plataforma. Esto se llevó a cabo, al pasar de la versión 6.0 a la versión 7.0 (llamada Visual Basic .NET 2002).

A continuación resumiremos brevemente los cambios que se añadieron en la versión Visual Basic .NET 2002:

- **Orientación a objetos:** en la versión 4.0, se crearon los *módulos de clase*, unos archivos individuales en los que se especificaban las propiedades y los métodos de una clase concreta. Era un avance, pero Visual Basic, seguía careciendo de las características básicas de la programación orientada a objetos, tales como la herencia y el polimorfismo.
- **Interfaces:** Este concepto ya existía en la versión 6.0, pero su utilidad era muy limitada, las interfaces eran inaccesibles para los usuarios de los objetos. En Visual Basic .NET las interfaces son construcciones de primer nivel.
- **Espacios de nombres:** o *manespace* no existía en la versión 6. Actualmente hay multitud de lenguajes que lo usan para estructurar sus tipos de datos.
- **Tipos de datos:** En Visual Basic .NET desaparece el tipo de dato *Variant*. Y se crea un tipo similar, el tipo *Object*. Este actúa como si se tratase de una clase y todos los demás tipos (incluso los tipos intrínsecos como *Integer* o *Char*), pueden ser tratados como derivados de él. Otro cambio impórtate, es el tipo *Integer*, en la versión 6, tiene 16bits, mientras que en la versión .NET tiene 32bits.
- **Control estructurado de excepciones:** Esta era una de las carencias más importante de la versión 6.

Novedades de Visual Basic 2005

En esta versión todos los aspectos relativos a la definición de clase, interfaces, espacios de nombres, propiedades, tipos de datos, delegados, etc., se mantienen como en la versión 2002. Lo que se ha hecho es agregar nuevas palabras clave y posibilidades que antes no existían, pero sin modificar ni eliminar aspectos introducidos con Visual Basic .NET. Estas mejoras son:

- **El espacio de nombres *My*:** Su importancia reside en reducir la cantidad de código que necesitamos escribir para llevar a cabo las tareas más cotidianas que son, las que más se repiten y en las que se emplea una importante cantidad de tiempo. En nuestra aplicación, se ha utilizado, por ejemplo, las siguientes operaciones:
 - Acceder a los recursos de nuestra aplicación, Cadenas de texto, Imágenes...
`My.Resources.CandenaConex`
 - Obtener una cadena de texto, con la ruta de nuestra aplicación
`My.Application.Info.DirectoryPath`
 - Guardar un texto al portapapeles
`My.Computer.Clipboard.SetText("Textol")`

- **Tipos de datos genéricos:** Se utilizan para englobar métodos genéricos a diferentes acciones. Aunque la utilidad de los datos genéricos es muy extensa, en nuestra aplicación hemos utilizado, sobretodo las listas de objetos.

```
Dim parametros As New List(Of SqlParameter)  
  
parametros.Add(BD.CrearParametro("@xml", SqlDbType.Xml, 0, True,  
_ ObtenerXML()))  
  
BD.EjecutarProcedimiento("CrearAutor", parametros)
```

- **Tipos de datos parciales:** facilita la división del código fuente con que se define un tipo, entre varios módulos. El entorno de desarrollo Microsoft Visual Studio, realiza de manera semitransparente para el usuario, con las clases formulario.

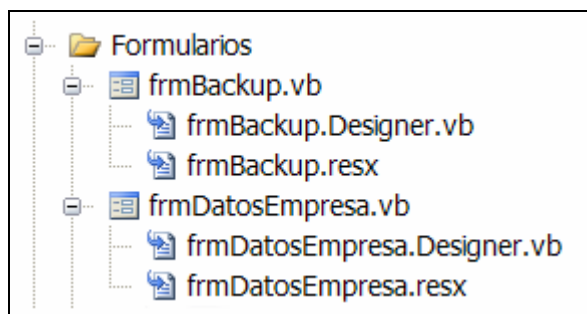


Figura 27: Ejemplo de Tipos de datos parciales

En el archivo *frmBackup.vb* nosotros introducimos nuestro código (variables miembro, métodos, eventos, etc.), mientras que la plataforma de desarrollo o IDE genera el archivo *frmBackup.Designer.vb*. En ambos ficheros, la definición de la clase *frmBackup* se realiza de la siguiente forma:

```
Partial Class frmBackup
```

- **Tipos de datos sin signo:** *UShort*, *UInteger* y *ULong*.
- **Operadores en tipos propios** o sobrecarga de operadores, nos permite operar sobre objetos, de un determinado tipo, utilizando los operadores estándar, como por ejemplo + o *. La cabecera de la definición de un operador sería similar a la siguiente:

```
Public Shared Operator +(Parámetros) As Tipo
```

- **Operadores de conversión:** mediante la definición de operadores a medida es posible facilitar también la conversión entre el nuevo tipo de dato.
- **Operadores lógicos:** en principio, un objeto de un tipo de usuario no puede ser empleado directamente para controlar un condicional, a menos, que usemos alguna de sus propiedades o métodos como valor para codificar una expresión relacional en el punto adecuado. Es decir, suponiendo que *T* es un objeto de *Tiempo* podemos hacer lo siguiente:

`If T.Horas = 0 Then ...` pero no lo siguiente: `If T Then ...`

Para que Visual Basic pueda determinar si una variable se evalúa como *True* o *False*, hay que agregar a la definición del tipo dos operadores: *IsTrue* e *IsFalse*.

- **Comentarios y documentación XML:** en la versión 2005 de Visual Basic existe la posibilidad de introducir comentarios que, siguiendo una estructura predefinida, se puede utilizar para generar posteriormente y de forma automática una documentación externa de las clases, métodos, etc. En el siguiente ejemplo se comente una variable:

```
''' <summary>
''' Variable miembro donde almacenaremos la conexión a la BD.
''' </summary>
''' <remarks></remarks>
Private conexion As SqlConnection
```

Al compilar el proyecto que contiene este código, Visual Basic producirá entonces un archivo XML con el mismo nombre del módulo de código. En él (ver figura XX) se recogen todos los comentarios de documentación.

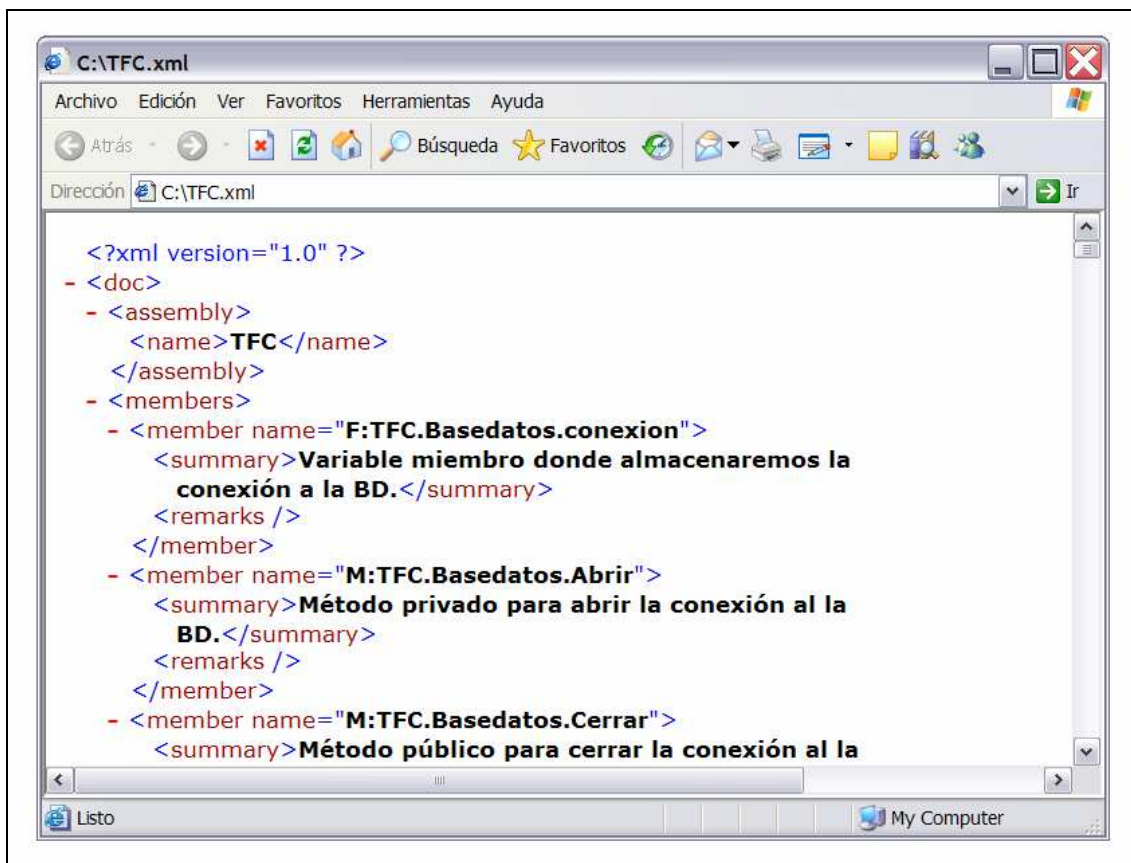


Figura 28: Archivo XML con los comentarios del proyecto.

9.1.3. ¿Por qué Visual Basic 2005?

Como he explicado anteriormente, Visual Basic ha ido evolucionado desde su creación (hace 15 años). Anteriormente había trabajado con Visual Basic 6.0, realizando pequeños proyectos. Esto hace que este familiarizado con su sintaxis.

Para mí suponía un reto, aprender a programar en Visual Basic .NET, ya que junto con C# (también de la tecnología .NET) y Java son lenguajes muy demandados en la actualidad.

Además, una de las ventajas de que programar en Visual Basic .NET, la sencillez y claridad de su código.

9.1.4. Crystal Reports

Para construir los informes, hemos utilizado Crystal Reports, un potente editor de informes que cuenta con una interfaz que permite elaborar, manipular e imprimir los informes. Esta herramienta está integrada en el paquete de *Visual Studio 2005 Professional Edition*.

Aunque no tenía experiencia en el uso de esta herramienta, sus asistentes, nos han facilitado un aprendizaje rápido, así como la posibilidad de crear informes, a los que después hemos ido añadiendo complejidad.

9.2. *Un poco de código*

En este apartado, realizaremos una descripción de la estructura interna de la aplicación. Explicando las cosas más reseñables de su código fuente.

9.2.1. Procedimientos Almacenados

Los procedimientos almacenados, son pequeños módulos (o funciones), que trabajan sobre las tablas de la base de datos. Además, están almacenados en la misma base de datos, agrupando de esta forma datos y operaciones.

Para entender mejor, el funcionamiento de los procedimientos almacenados, a continuación describiremos cuatro de los más representativos:

- AutenticarUsuario
- CrearAutor
- EliminarAutor
- ListarAlmacen

AutenticarUsuario

Es el procedimiento guardado que utiliza el sistema para autenticar a un usuario. Los parámetros de entrada son el usuario (o *login*) y la contraseña. Como parámetro de salida tenemos la variable *nombre*, que en el caso de que la autenticación se realice correctamente, el procedimiento devolverá una cadena de caracteres con el nombre y apellidos del usuario.

Código del procedimiento:

```
CREATE PROCEDURE AutenticarUsuario
(
    @usuario varchar(50),
    @password nchar(10),
    @nombre varchar(50) OUTPUT
)
AS
    DECLARE @idUsuario int

    SELECT @idUsuario=idUsuario, @nombre=nombre
    FROM Usuarios
    WHERE usuario=@usuario AND password=@password

    IF @@ROWCOUNT < 1
        SELECT @idUsuario = 0

    RETURN @idUsuario
```

Comentarios adicionales sobre el código:

- La variable *@@Rowcount*, contiene el número de registros generados, de la ejecución de la consulta *Select* anterior.
- El modificador *OUTPUT* indica que el parámetro es de salida.

CrearAutor

Este es el procedimiento almacenado, es el que usamos para añadir un nuevo autor a la base de datos. La característica principal, es el uso de un único parámetro de tipo XML. Esto es así para ahorrar la complicación que supone, trabajar con tantos parámetros, como atributos tenga la tabla.

Veamos un ejemplo, de cómo sería un el contenido, del parámetro @xml:

```
<Autor nombre='Ricardo'
  apellidos='Garcia Coscolla'
  DNI='73129922X'
  direccion='c/ del Arco nº 32'
  codigoPostal='22561'
  localidad='Tamarit'
  provincia='Huesca'
  pais='España'
  telefono='974998877'
  movil='666998877'
  fax='974998871'
  email='ricardogc@yahoo.es'
  CCC=''>
</Autor>
```

Código del procedimiento:

```
CREATE PROCEDURE CrearAutor
(
  @xml xml
)
AS
  DECLARE @idoc int      --xml doc
  DECLARE @idAutor int

  -- Interpretamos el documento xml
  EXEC sp_xml_preparedocument @idoc output, @xml

  SET NOCOUNT ON
  DECLARE @CurrentError int

  -- Añadimos un nuevo autor
  INSERT INTO Autores
  SELECT nombre, apellidos, DNI, direccion, codigoPostal,
  localidad, provincia, pais, telefono, movil, fax, email, CCC
  FROM OpenXML (@idoc, '/Autor')
  WITH Autores

  -- Guardamos el identificador del autor.
  SET @idAutor = @@Identity

  -- Comprovamos posibles errores
  SELECT @CurrentError = @@Error

  IF @CurrentError != 0
  BEGIN
    GOTO ERROR_HANDLER
  END
```

```
SET NOCOUNT OFF

-- Hemos terminado con el documento xml
EXEC sp_xml_removedocument @idoc

-- Retornamos el @idAutor para indicar OK
RETURN @idAutor

ERROR_HANDLER:
    SET NOCOUNT OFF
    RETURN 0

RETURN
```

Comentarios adicionales sobre el código:

- “*EXEC sp_xml_preparedocument @idoc output, @xml*” es la rutina encargada de interpretar el contenido, del parámetro *@xml* y almacenarlo en la variable *@idoc*.
- La variable *@@Identity*, almacena el identificador, de la última operación (de tipo *insert*, *update* o *delete*) realizada.
- *@@Error* nos retorna los posibles errores que se han generado al realizar alguna operación.
- Una vez hemos terminado de trabajar con el documento XML, hay que liberarlo de la memoria, para ello utilizamos la siguiente expresión: “*EXEC sp_xml_removedocument @idoc*”

EliminarAutor

Este procedimiento almacenado, elimina un autor de la base de datos. Como único parámetro tenemos el *@idAutor* del autor que deseamos eliminar.

Código del procedimiento:

```
CREATE PROCEDURE EliminarAutor
(
    @idAutor int
)
AS
    DELETE FROM Autores WHERE idAutor=@idAutor

    RETURN 1
```

ListarAlmacen

Como podemos observar, este procedimiento carece de parámetros, tanto de entrada como de salida. Esto es así, por el procedimiento almacenado, solamente se una para, obtener datos, de la base de datos, sin realizar ningún tipo de selección externa.

Código del procedimiento:

```
CREATE PROCEDURE ListarAlmacen
AS
    SELECT dbo.Libros.ISBN,
           dbo.Libros.titulo,
           dbo.Libros.fechaPublicacion,
           dbo.Autores.nombre,
           dbo.Autores.apellidos,
           dbo.Libros.tirada,
           dbo.Libros.ejemplaresEditados,
           dbo.Almacenes.stock,
           dbo.Libros.stockMinAlmacen,
           dbo.Libros.pvpSinIva,
           dbo.Libros.pesoLibro
    FROM   dbo.Almacenes
           INNER JOIN dbo.Libros
                   ON dbo.Almacenes.refLibro = dbo.Libros.idLibro
           INNER JOIN dbo.Autores
                   ON dbo.Libros.refAutor = dbo.Autores.idAutor
WHERE (dbo.Almacenes.refDistribuidor = 1)
RETURN
```

9.2.2. Visual Basic

En este apartado, explicaremos los aspectos más significativos del código de la aplicación:

Estructura de la aplicación

Como hemos comentado anteriormente, para interactuar con las tablas de la base de datos utilizamos procedimientos almacenados. Esto hace que nuestros formularios, casi no dispongan de lógica de la negocio. En otras palabras, su función es la de mostrar los datos, validar las entradas y enviar los datos necesarios a la base de datos, para que esta (por medio de los procedimientos almacenados), se guarde la información de forma correcta. Para poder establecer la comunicación entre la base de datos y los formularios, hemos creado la clase *basedatos* (ver anexo II).

Herencia visual

Para el realizar el diseño y la implementación de los formularios (o pantallas) más parecidas, se ha recurrido a la *herencia visual*. Esta técnica consiste en crear un formulario base con aquellos controles, métodos, propiedades... y hacer que los nuevos formularios sean agregaciones del formulario base. De esta forma se consigue un mismo criterio, tanto en el diseño de los formularios, como en la forma implementación de estos.

Como podemos observar en la siguiente figura, los formularios *frmGestionLibros*, *frmGestionAutores* y *frmGestiónClientes* son hijos de *frmListadoBase*. Y los formularios *frmFichaLibro*, *frmFichaAutor* y *frmFichaDistribuidorCliente*, son agregaciones de *frmFichaBase*.

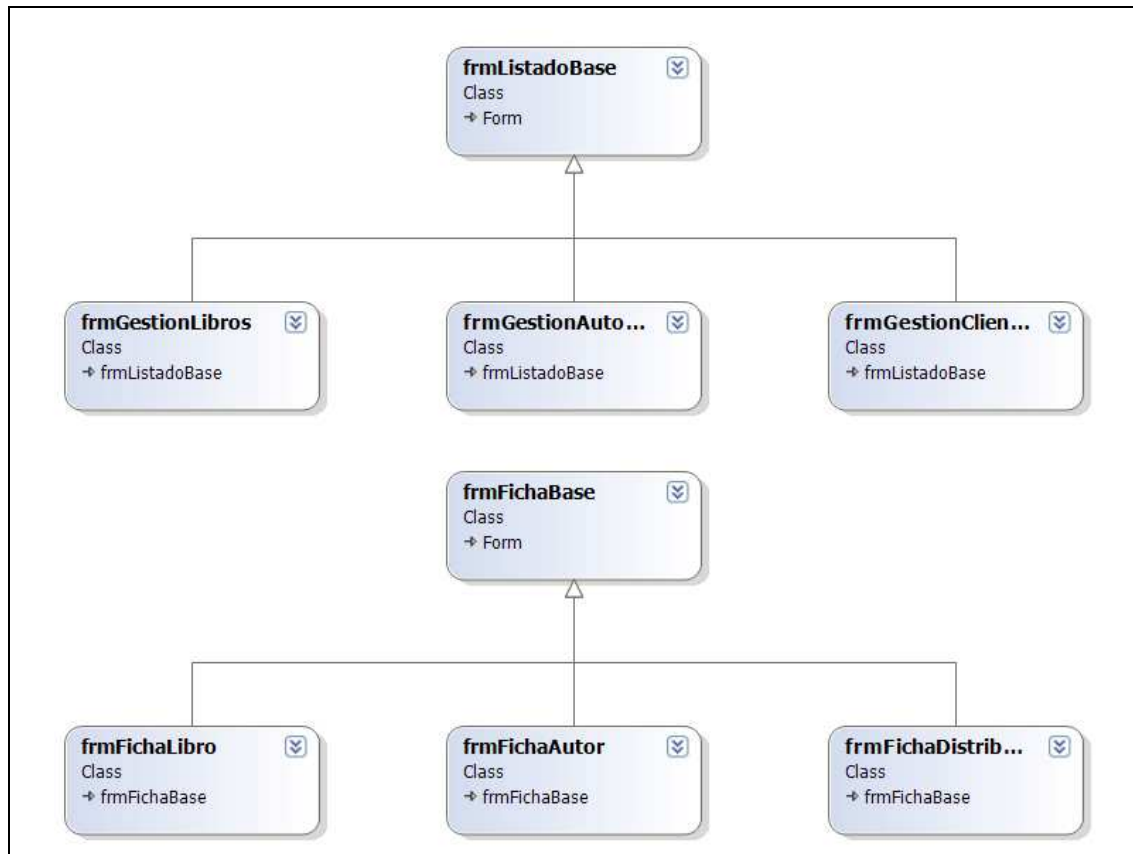


Figura 29: Diagrama de clases “Herencia visual”

A continuación, veremos como es el diseño de *frmListadoBase*:

Interfaz de usuario de la ventana **frmListadoBase**. La ventana tiene un título "Titulo Listado" y una barra de herramientas con los botones: **Añadir**, **Modificar**, **Eliminar** y **Imprimir**. El área principal es un espacio gris vacío para la lista. En la esquina inferior derecha hay un botón **Cerrar** con un icono de una X roja.

Figura 30: Diseño de formulario “frmListadoBase”

En el código del formulario, hemos creado ocho métodos que son los únicos que deberán ser sobrescritos en las clases hijo. *frmListadoBase* controla los eventos del formulario y los asocia a estos métodos:

```
#Region "Métodos a sobrescribir"
    Protected Overridable Function CargarDatos() As DataTable
    End Function

    Protected Overridable Sub OcultarColumnasDataGridView()
    End Sub

    Protected Overridable Sub MaquetarColumnasDataGridView()
    End Sub

    Protected Overridable Sub RenombrarEtiquetas()
    End Sub

    Protected Overridable Function AñadirFila(ByVal fila As
DataGridViewRow) As Boolean
    End Function

    Protected Overridable Function ModificarFila(ByVal fila As
DataGridViewRow) As Boolean
    End Function

    Protected Overridable Function EliminarFila() As Boolean
    End Function

    Protected Overridable Sub ImprimirListado()
    End Sub
#End Region
```

Comentarios:

- El modificador *Overridable*, indica que el procedimiento puede sobrescribirse.
- El uso de las etiquetas *#Region* “Métodos a sobrescribir” y *#End Region*, nos ayuda delimitar y agrupar las partes del código.

Copias de seguridad

Utilizando el método *Backup* de la clase *Basedatos* (ver anexo II), el sistema realiza copias de seguridad de la base de datos. Estas copias a su vez son las que utiliza el sistema para restaurar la base de datos, en caso que se necesario. Esa copia de seguridad se podrá volver a restaurar en ese mismo equipo o en otro diferente, por que el formado de esta es el mismo que el que usa SQL Server 2005, para crear sus copias de seguridad. La secuencia de comando que usa el método *Backup* de la clase *Basedatos* es la siguiente:

```
"BACKUP DATABASE [" & rutaBD & "] TO DISK = N'" & rutaCopiaSeg & "'
WITH NOFORMAT, NOINIT, NAME = N'C:\GESTEDI\GESTEDI.MDF-Completa Base
de datos Copia de seguridad', SKIP, NOREWIND, NOUNLOAD, STATS = 10"
```

Donde *rutaBD* es la ubicación del archivo *gestedi.mdf* y *rutaCopiaSeg* es la ubicación del archivo, donde almacenaremos la copia de seguridad.

10. Pruebas y validación

Antes de la finalización y venta de nuestro producto, este tiene que superar una serie de pruebas de validación, para asegurar que nuestro producto funciona correctamente i no existe ningún conflicto.

Un producto no puede lanzarse al mercado, si que este haya superado unas pruebas que certifiquen que funciona correctamente. Estas pruebas son las siguientes:

- Pruebas de unidad.
- Pruebas de integración.
- Pruebas de validación.
- Pruebas de sistema.

10.1. Pruebas de unidad

La prueba de unidad se basa en la verificación de cada modulo del sistema, de forma independiente, se probarán los principales caminos de control y se aplicarán las técnicas de caja blanca. Estas consisten en probar las interfaces, las estructuras de datos locales, los diferentes caminos independientes, los tratamientos de errores y las condiciones límites.

Las pruebas de caja blanca son un camino minucioso de los detalles. En esta prueba se comprueba los caminos lógicos del software proponiendo casos de prueba que ejecuten conjuntos de caminos específicos de bucles y condiciones. El inconveniente es que realizar una prueba exhaustiva de bucles y condiciones presenta problemas logísticos.

Para llevar a cabo dicha prueba se han realizado las siguientes tareas:

- Se han ejecutado todos los módulos para ver si realmente tenían el comportamiento deseado.
- Se han analizado todos los caminos independientes asegurando la ejecución de las sentencias de cada módulo.
- Se ha probado la aplicación en condiciones límites, ya que el software normalmente falla en esas situaciones.
- Se han examinado las estructuras de datos, verificando que en todo momento, se mantienen íntegras.

10.2. Pruebas de integración

Esta prueba es una técnica sistemática para construir la estructura del programa mientras que, al mismo tiempo, se llevan a cabo pruebas para detectar errores asociados a la interacción. El objetivo es coger los módulos probados en unidad y construir una estructura de programa que esté de acuerdo con lo que dicta el diseño.

Este es un tipo de prueba muy importante, ya que puede suceder que módulos de la aplicación tengan efectos negativos sobre otros, o que en la combinación de subfunciones no se obtenga la función principal que se espera.

Durante la implementación de la aplicación, la integración se ha realizado de forma incremental.

Esta aplicación se ha desarrollado probando por separado los módulos y añadiendo de nuevos a medida que estos módulos estaban bien contruidos.

10.3 Pruebas de validación

Esta prueba se realiza una vez finalizada toda la integración del sistema, el software está completamente ensamblado como un paquete, se han encontrado y corregido los errores de interfaz y puede comenzar la serie final de pruebas de software, la prueba de validación.

La validación consiste en probar que la aplicación funciona de acuerdo con lo que se había especificado en los requisitos. Para realizar esta prueba se llevan a cabo las pruebas *Alfa* y *Beta*. La prueba *Alfa* es conducida por el cliente en el lugar de desarrollo y se utilizará el software de manera natural con el encargado del desarrollo del producto. La prueba *Beta* se lleva a cabo en el lugar del cliente y por el usuario final, el desarrollador no esta presente.

La interacción de cliente con la aplicación ha sido satisfactoria, en parte gracias a que en la fase de construcción, durante la implementación hubo muchos intercambios de opiniones con el cliente.

10.4 Pruebas de sistema

Estas pruebas están formadas por una serie de pruebas diferentes, con el objetivo de probar que todos los elementos del sistema informático (software y hardware), se han integrado correctamente y realizan la función esperada.

Las pruebas de las que se componen la prueba de sistema son:

- Prueba de recuperación
- Prueba de seguridad
- Prueba de resistencia
- Prueba de rendimiento

10.4.1. Prueba de recuperación

En esta prueba se fuerza la caída del sistema de diversas formas y verifica que la recuperación se realiza de forma correcta.

Para realizar esta prueba, se provocó la caída del sistema cuando la aplicación estaba en funcionamiento. Inmediatamente, se comprobó que el funcionamiento de la aplicación era el correcto.

10.4.2. Prueba de seguridad

Esta prueba verifica que los mecanismos de protección incorporados al sistema, los protegerán de posibles accesos no deseados.

En nuestro caso, se comprueba que cualquier persona que entre sin permiso no puede acceder a la aplicación, ya que necesita una cuenta de usuario y una contraseña. Estas solamente las puede facilitar el administrador de la aplicación.

10.4.3. Prueba de resistencia

Esta prueba se encarga de llevar la aplicación a situaciones extremas y anormales. Se ejecuta el sistema de forma que utilice recursos en cantidades, frecuencia o volúmenes anormales.

En el caso de nuestra aplicación, se ha sometido a la aplicación de consultas en la base de datos en frecuencias y volúmenes de datos superiores a los que se utilizarán.

10.4.4. Prueba de rendimiento

La prueba de rendimiento está diseñada para probar el rendimiento del software, en tiempo de ejecución de un sistema integrado.

En nuestra aplicación, se ha probado la aplicación en diferentes equipos, entre ellos en el equipo donde se utilizará y los resultados han sido satisfactorios.

11. Conclusiones y ampliaciones

11.1 Conclusiones

El objetivo básico del proyecto, el cual se ha conseguido, era proporcionar a empresarios de pequeña y mediana empresa (mas concretamente al sector editorial), una herramienta informática que les ayudará en la gestión de su empresa.

El hecho de escoger varios modelos de desarrollo de software, modelo secuencial, modelo incremental y construcción de prototipos, ha facilitado el diseño y desarrollo de la aplicación y una vez concluidas las tareas del proyecto puedo afirmar que se han satisfecho todos los requisitos iniciales del proyecto y que nos encontramos ante una primera versión de producto lista para su distribución.

Durante la elaboración del proyecto he encontrado bastantes dificultades, sobretodo en los primeros pasos, la recogida de requisitos y entender la visión global de la empresa. También se me plantearon, bastantes interrogantes en la fase de diseño de la aplicación. Y al principio, me costo adaptarme a la manera de programar en Visual Basic .NET. Superar estas dificultades ha supuesto un ejercicio didáctico excelente.

La elaboración de un proyecto de estas características me ha permitido adquirir nuevos conocimientos, así como tener contacto con el mundo laboral en el desarrollo de aplicaciones informáticas. Además he podido poner en práctica bastantes conocimientos adquiridos a lo largo de la carrera.

11.2. Ampliaciones

Una aplicación de estas características está siempre abierta a nuevas ampliaciones y mejoras funciones. En el mismo instante que se finaliza una versión, ya surgen ideas y ampliaciones futuras.

Algunas ampliaciones futuras podrían ser:

- Que la aplicación pudiera conectarse con alguna herramienta contable, de las que ya existen en el mercado.
- Actualmente, la función de las cuentas de usuario, solamente sirven para controlar el acceso a la aplicación. En un futuro sería interesante añadir una gestión de roles, para restringir el acceso de algunos usuarios, a ciertas pantallas.
- Controlar la ubicación de los libros en el almacén.
- Añadir un modulo que gestione los derechos de autor.

12. Bibliografía

PRESSMAN, R. *Ingeniería de Software. Un enfoque práctico*. ED. McGraw-Hill. Segunda edición, Agosto 1992.

CHARTE, F. *Visual Basic 2005*. ED. Ediciones Anaya Multimedia, Febrero 2005.

SERRANO, J. *Manual avanzado de Visual Basic 2005*. ED. Ediciones Anaya Multimedia, Febrero 2006.

PETROUTSOS, E. *La Biblia de Visual Basic.net* ED. Ediciones Anaya Multimedia, Marzo 2003.

Apuntes de asignaturas:

GIMENO, J.M. *Aplicació a l'Enginyeria del Software*. 2005

LORÉS, J. *Labosoft*. 2004

OLIVA, M. *Aplicació a Bases de Dades*. 2004

Recursos y enlaces Web:

MSDN de Microsoft

<http://www.msdn.microsoft.com/> (21/03/2007)

SDK de Microsoft .NET Framework 2.0:

<http://www.msdn.microsoft.com/netframework/downloads/updates/> (21/03/2007)

Los 101 ejemplos de Visual Basic 2005:

<http://www.msdn.microsoft.com/vbasic/downloads/101samples/> (11/04/2007)

Libro electrónico gratuito en inglés, sobre la migración de Visual Basic 6 a Visual Basic.NET

<http://www.msdn.microsoft.com/vbrun/staythepath/additionalresources/upgradingvb6/> (09/03/2007)

El Guille, Web de programación:

<http://www.elguille.info/> (01/08/2007)

Desarrollador 5 Estrellas

<http://www.mslatam.com/latam/msdn/comunidad/dce2005/> (01/08/2007)

ANEXO I: Manual de usuario

Índice del manual

Índice de figuras	80
1. Introducción.....	81
1. Introducción.....	81
2. Pantalla de acceso a la aplicación.....	82
3. Pantalla Principal.....	83
3.1 Menú principal de la aplicación.....	83
3.1.1 Menú de Pantallas.....	84
3.1.2 Menú de Listados	85
3.1.3 Menú Ver.....	85
3.1.4 Menú Herramientas.	86
3.1.5 Menú de Ventanas	88
3.2 Barra de herramientas del programa.....	89
3.3 Barra de estado del programa.	89
4. Pantallas.....	90
4.1 Pantalla de Autores.....	90
4.1.1 Añadir un autor	91
4.1.2 Modificar un autor	92
4.1.3 Eliminar un autor	93
4.1.4 Listar los autores.....	94
4.2 Pantalla de Sellos, Estilos y Colecciones	95
4.2.1 Añadir un nuevo elemento a la lista	96
4.2.2 Modificar un elemento	97
4.2.3 Eliminar un elemento	97
4.3 Pantalla de Libros	98
4.3.1 Añadir un libro	99
4.3.2 Modificar un libro.....	100
4.3.3 Eliminar un libro.....	100
4.3.4 Listar los libros	101
4.4 Pantalla de Almacén	102
4.5 Pantalla de Clientes	104
4.6 Pantalla de Distribuidores.....	106
4.7 Pantalla de Movimientos	109
4.7.1 Pestaña “Ventas y Movimientos entre distribuidores”	109
4.7.2 Pestaña “Rectificación de una venta”	111
4.8 Pantalla de Facturas	112
4.8.1 Pestaña Listado de facturas.....	112
4.8.2 Pestaña Facturar a Distribuidores	114
4.8.3 Pestaña Facturar a Clientes.....	116
5. Otros Listados.....	118

Índice de figuras

Figura 1: Acceso.....	82
Figura 2: Informa al usuario que el nombre de usuario o la contraseña son erróneos. ..	82
Figura 3: Pantalla Principal.....	83
Figura 4: Menú de Pantallas.....	84
Figura 5: Menú de Listados.....	85
Figura 6: Menú Ver.....	85
Figura 7: Menú Herramientas.....	86
Figura 8: Pantalla datos de la empresa.....	86
Figura 9: Pantalla Copias de seguridad.....	87
Figura 10: Restaurar la base de datos.....	87
Figura 11: Ejemplo de información rápida.....	87
Figura 12: Menú Ventanas.....	88
Figura 13: Barra de herramientas.....	89
Figura 14: Barra de estado.....	89
Figura 15: Ejemplo de información en la barra de estado.....	89
Figura 16: Pantalla de Autores.....	90
Figura 17: Nuevo Autor.....	91
Figura 18: Modificar Autor.....	92
Figura 19: Eliminar un autor.....	93
Figura 20: Error al eliminar un autor.....	93
Figura 21: Listado de autores.....	94
Figura 22: Pantalla de sellos, estilos y colecciones.....	95
Figura 23: Menú contextual de la lista de estilos.....	96
Figura 24: Pantalla de nuevo Sello.....	96
Figura 25: Pantalla de modificar colección.....	97
Figura 26: Eliminar un sello.....	97
Figura 27: Pantalla de Libros.....	98
Figura 28: Nuevo Libro.....	99
Figura 29: Lista de Autores.....	100
Figura 30: Listado de Libros.....	101
Figura 31: Pantalla de Almacén.....	102
Figura 32: Pantalla de Almacén (Libros con Stock inferior al Stock Min.).....	103
Figura 33: Pantalla Clientes.....	104
Figura 34: Listado de Clientes.....	105
Figura 35: Ficha de un Distribuidor.....	106
Figura 36: Pantalla de Distribuidores.....	107
Figura 37: Listado de Distribuidores.....	108
Figura 38: Pantalla Movimientos (Pestaña “Ventas y Movimientos”).....	109
Figura 39: Pantalla Movimientos (Opción “Venta”).....	110
Figura 40: Pantalla Movimientos (Pestaña “Rectificación de una venta”).....	111
Figura 41: Pantalla Facturas.....	112
Figura 42: Pantalla Imprimir Facturas.....	113
Figura 43: Listado Facturas.....	113
Figura 44: Pestaña Facturar a Distribuidores.....	114
Figura 45: Pantalla Vista preliminar de la Factura.....	115
Figura 46: Pantalla Facturar a Clientes.....	116
Figura 47: Imprimir movimientos.....	118

1. Introducción

GESTEDI es una aplicación que gestiona la logística de una pequeña editorial de libros. Realizando un mantenimiento riguroso de los autores, libros, clientes, distribuidores... También podremos realizar la facturación y listados que nos facilitaran las tareas.

A la hora de diseñar esta aplicación, se ha buscado por encima de todo, maximizar la sencillez. Evitando la sobrecarga de botones e información, en las distintas pantallas de esta.

A continuación, realizaremos un amplio recorrido por la aplicación explicando como y para que sirven las diferentes unidades de esta.

Nota: A lo largo de este documento, el lector encontrará entre paréntesis anotaciones de este tipo:

(o teclear “Alt” + “A”)

(o “Alt” + “C”)

Estas anotaciones son las teclas rápidas para botones, menús... pulsándolas, evitaremos tener que utilizar el ratón, para navegar por las pantallas.

2. Pantalla de acceso a la aplicación

Cuando ejecutamos el programa lo primero que visualizaremos es la pantalla de acceso a la aplicación, la podemos ver en la figura 1.



Figura 31: Acceso...

En esta pantalla, tenemos que introducir el nombre de usuario y su contraseña, con el cual accedemos a la aplicación. Luego tenemos que pulsar el botón Aceptar (o teclear Alt + A), para entrar a la aplicación. Si el nombre de usuario o la contraseña son incorrectos el programa nos lo informará (ver figura 2). En ese caso, volveremos a introducir el nombre de usuario y su contraseña. Si el problema persiste, tendremos que ponernos en contacto con el administrador de la aplicación, para que nos describa el usuario y password.

Si pulamos el botón de Cancelar (o Alt + C), la aplicación se cerrará, sin que aparezca la pantalla principal.

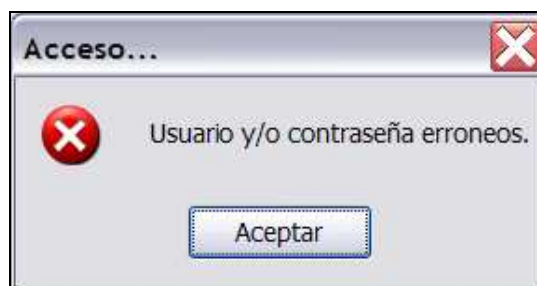


Figura 32: Informa al usuario que el nombre de usuario o la contraseña son erróneos.

3. Pantalla Principal

Una vez realizada la autenticación (mediante la pantalla de acceso), se visualizará la pantalla principal del programa (ver figura 3).

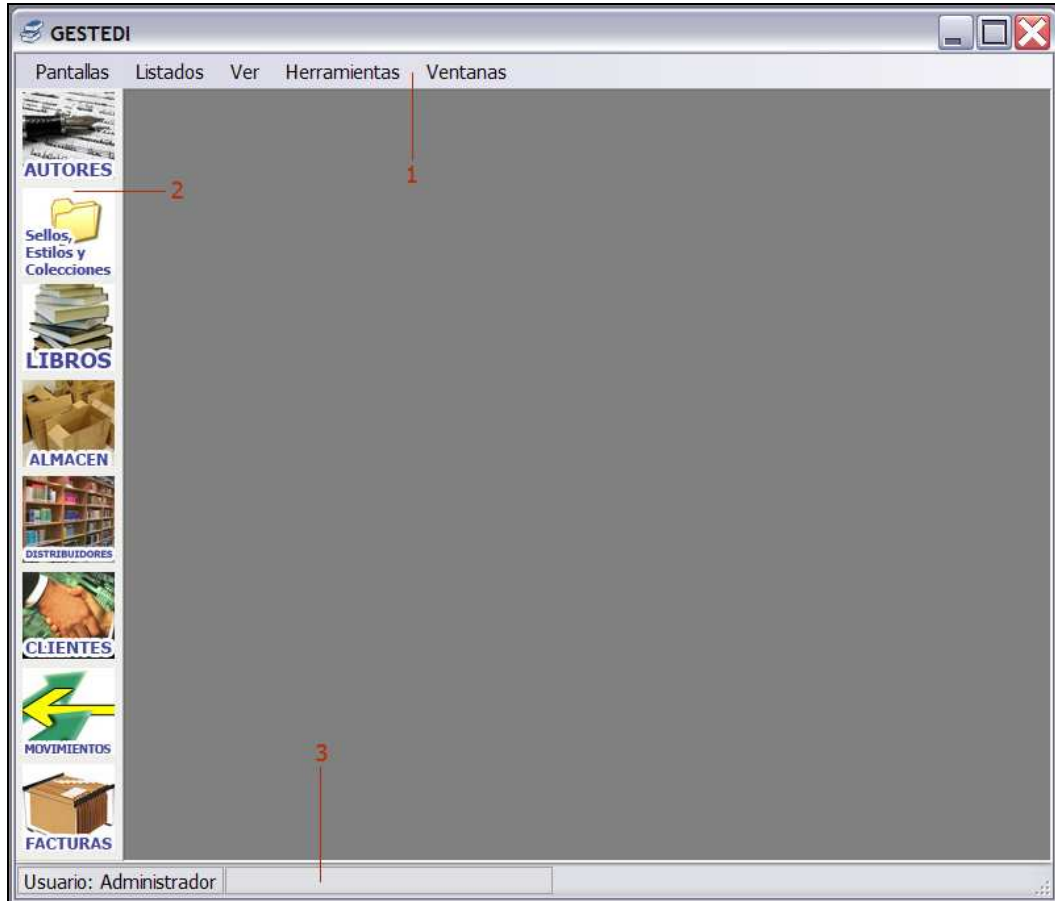


Figura 33: Pantalla Principal.

Dentro de la pantalla principal, podemos distinguir tres partes (señaladas en la figura 3):

1. Menú principal de la aplicación.
2. Barra de herramientas.
3. Barra de Estado.

3.1 Menú principal de la aplicación

Por medio del menú principal, podremos acceder a todas las opciones de la aplicación. El menú esta agrupado de la siguiente forma:

1. Pantallas.
2. Listados.
3. Ver.
4. Herramientas.
5. Ventanas.

3.1.1 Menú de Pantallas

Si clicamos en el menú de pantallas (o “Alt” + “P”), nos aparece el siguiente submenu (ver figura 4). Desde este, podemos abrir las siguientes pantallas:

- Autores (“Alt” + “A”)
- Sellos, Estilos y Colecciones (“Alt” + “S”)
- Libros (“Alt” + “L”)
- Almacén (“Alt” + “M”)
- Clientes (“Alt” + “C”)
- Movimientos (“Alt” + “V”)
- Distribuidores (“Alt” + “D”)
- Facturas (“Alt” + “F”)
- Usuarios (“Alt” + “U”), esta última, sólo se nos mostrará, si accedemos a la aplicación con el usuario “Administrador” y sirve para gestionar las cuentas de usuario.



Figura 34: Menú de Pantallas.

3.1.2 Menú de Listados

En el menú de Listados (o “Alt” + “L”), nos aparece el siguiente submenú (ver figura 5).



Figura 35: Menú de Listados

3.1.3 Menú Ver

Este menú nos servirá para ocultar o mostrar la barra de herramientas y la barra de estado. Por defecto, se muestran ambas (Ver figura 6), pero el usuario podrá ocultarlas siempre que lo estime oportuno.

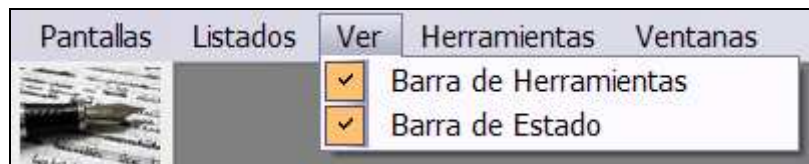


Figura 36: Menú Ver

3.1.4 Menú Herramientas.

El menú de Herramientas (ver figura 7) esta separado en dos partes, herramientas propias de la aplicación (Datos de la empresa, copias de seguridad, restaurar la base de datos y mostrar información rápida) y herramientas de Windows (la calculadora y el bloc de notas). Las herramientas de Windows se han incluido para facilitar el acceso a estas.

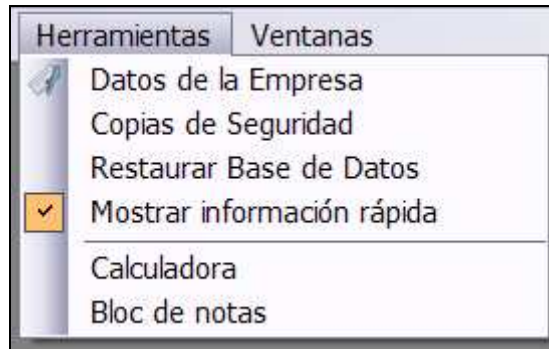


Figura 37: Menú Herramientas

En la pantalla de Datos de la empresa (figura 8), podemos modificar los datos de nuestra empresa. Si pulsamos el botón de cerrar, la pantalla se cerrará sin guardar los cambios. Pero si por el contrario, queremos guardar algún cambio, tenemos que pulsar el botón de guardar, la pantalla se cerrará y el nuevo dato será almacenado.

A screenshot of a form titled 'Datos de la Empresa' (Company Data) with a close button (X) in the top right corner. The form contains several input fields for company information: 'Nombre' (Name) with 'Mi Editorial', 'NIF' with '73204825-X', 'Dirección' (Address) with 'Calle mayor nº 5', 'Código Postal' (Postal Code) with '22000', 'Localidad' (Municipality) with 'Huesca', 'Provincia' (Province) with 'Huesca', 'País' (Country) with 'España', 'Teléfono' (Phone) with '974421919', 'Tel. móvil' (Mobile phone) with '674421919', 'Fax' with '974421919', and 'Correo Electrónico' (Email) with 'info@miempresa.com'. At the bottom right, there are two buttons: 'Guardar' (Save) with a green disk icon and 'Cerrar' (Close) with a red X icon.

Figura 38: Pantalla datos de la empresa

El programa realiza copias de seguridad cada vez que se cierra. Pero nosotros podemos realizarlas, siempre que queramos utilizando la opción “Copias de seguridad”. Nos aparecerá la siguiente pantalla, donde podremos la ruta del archivo a guardar:



Figura 39: Pantalla Copias de seguridad

También se puede restaurar la base de datos, utilizando un archivo copia de seguridad. Para ello, pulsaremos la opción del menú, Restaurar la base de datos y nos aparecerá la siguiente pantalla:



Figura 40: Restaurar la base de datos

Utilizando esta pantalla, seleccionaremos el archivo copia de seguridad que queremos usar para la restauración.

Otra opción importante del menú Herramientas es la de mostrar información rápida. Si tenemos esta opción habilitada el programa nos mostrará una guía rápida en aquellas pantallas con más complicadas.

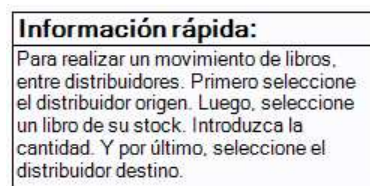


Figura 41: Ejemplo de información rápida

3.1.5 Menú de Ventanas

Este menú es de gran utilidad, para ordenar y cerrar nuestras pantallas y listados.

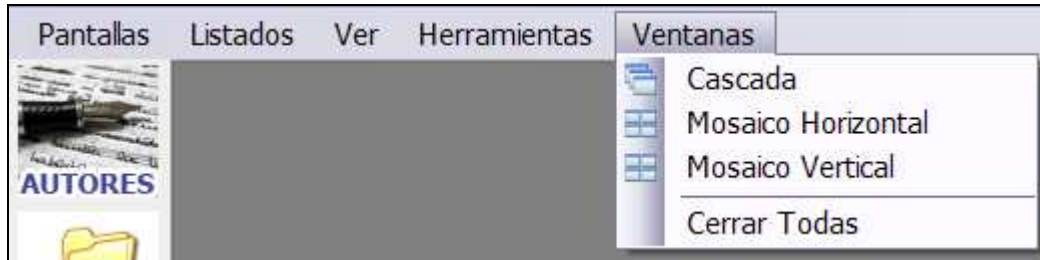


Figura 42: Menú Ventanas

Consta de las opciones siguientes:

- Cascada, nos pondrá todas las ventanas en cascada. Esta opción es muy útil, si trabajamos con varias ventanas y queremos ordenarlas.
- Mosaico Horizontal, esta opción nos pondrá todas las ventanas en mosaico horizontal. Resulta muy útil, cuando trabajamos con dos ventanas y queremos ver la información de ambas.
- Mosaico Vertical, esta opción nos pondrá todas las ventanas en cascada. Resulta muy útil, cuando trabajamos con dos ventanas y queremos ver la información de ambas.
- Cerrar Todas, nos cierra todas las pantallas o listados que tengamos abiertos.

3.2 Barra de herramientas del programa.

La barra de herramientas del programa (figura 13), es un acceso rápido y más accesible, a las pantallas de la aplicación. Por ejemplo, para acceder a la pantalla de Autores podemos hacerlo de dos formas:

- Realizando un clic en el botón de Autores de la barra de herramientas.
- Haciendo clic al menú “Pantallas” y luego a la opción “Autores”.



Figura 43: Barra de herramientas

Como hemos explicado anteriormente, esta barra se puede ocultar o mostrar, ultimando la opción “Barra de Herramientas” del menú “Ver”.

3.3 Barra de estado del programa.

La barra de estado (figura 14), nos muestra información útil de programa. Dicha información esta dividida en tres partes:



Figura 44: Barra de estado

1. Usuario del programa. Este aparato, nos muestra el nombre del usuario que accede a la aplicación
2. Ventana activa. Nos muestra el nombre la ventana activa.
3. Información relativa a la ventana activa. En esta sección, se muestran los mensajes informativos relativos a la pantalla activa. Estos son, en su mayor parte, mensaje que nos informan que el alta, baja o modificación, que acabamos de ejecutar, se ha realizado correctamente. Estos mensajes permanecen visibles durante 5 segundos.

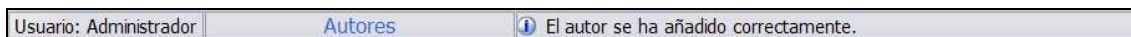


Figura 45: Ejemplo de información en la barra de estado.

4. Pantallas

4.1 Pantalla de Autores

Esta pantalla, es la encargada del mantenimiento de los autores. La podemos separar en varias partes (ver figura 16):

1. Es una pequeña barra de herramientas, con cuatro botones, para añadir, modificar, eliminar e imprimir. Podemos cambiar su posición, si nos resulta mas como, posesionándola al fondo o a ambos lados de la pantalla. Esta barra también la podemos encontrar en las pantallas de libros, clientes y distribuidores.
2. Listado corto de los autores, en el podremos observar los datos más significativos de cada autor. El ancho de sus columnas es dimensionable y también podemos cambiar el orden de dichas columnas.
3. Botón de cerrar.

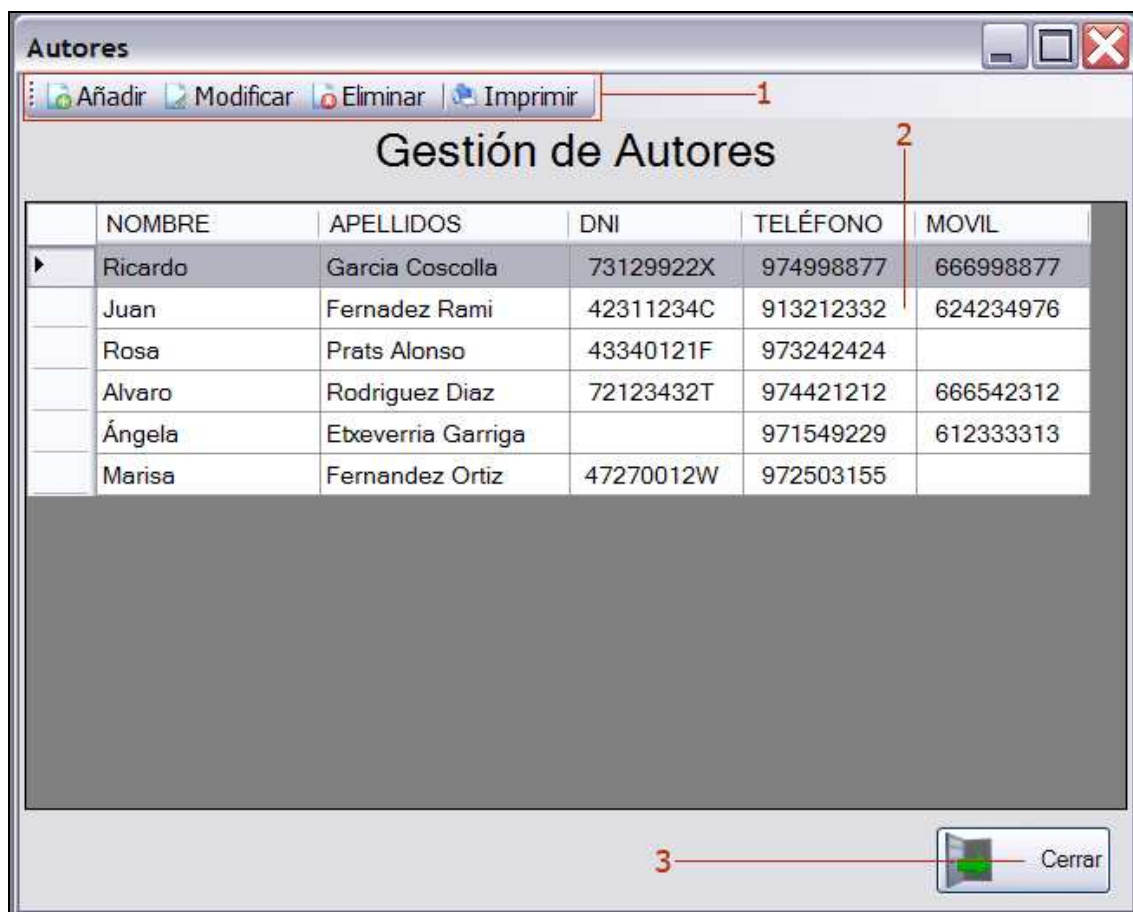


Figura 46: Pantalla de Autores.

Desde esta pantalla podremos añadir un nuevo autor, modificar los de un autor seleccionado y eliminar un autor (sin libros asignados). También se puede visualizar y/o imprimir un listado, con los datos de los autores, desde la pantalla de Autores (figura 17). Todas estas acciones se realizan por medio de la barra de herramientas de la pantalla.

4.1.1 Añadir un autor

Para añadir un nuevo autor, tenemos que presionar el botón de “Añadir”. Automáticamente nos aparece una ventana donde introduciremos todos los datos del autor. Una vez introducidos los datos, clicaremos el botón de guardar (o “Alt” + “G”) y la pantalla “Nuevo Autor” se cerrará, guardando los datos introducidos en la base de datos. Si por el contrario, no guardar los datos, presionaremos el botón de cerrar (o las teclas “Alt” + “C”) y la pantalla se cerrará sin guardar ningún dato del autor.

The screenshot shows the 'Autores' application window. In the top toolbar, the 'Añadir' button is highlighted with an arrow. A 'Nuevo Autor' dialog box is open, displaying a list of authors on the left and a form for entering new author data on the right. The form includes fields for personal and contact information. The 'Guardar' and 'Cerrar' buttons are at the bottom right of the dialog.

Figura 47: Nuevo Autor

4.1.2 Modificar un autor

Para acceder a la ventana con todos los datos del autor y modificarlos tenemos dos opciones (ver figura 18):

1. Clicar el botón de Modificar.
2. Hacer doble clic en la fila que queramos modificar. Esta opción es muy rápida y nos sirve de gran ayuda para consultar todos los datos de un autor.

Una vez en la pantalla de “Modificar Autor”, sólo tenemos que modificar los datos que precisos y presionar el botón de guardar (o las teclas “Alt” + “G”). La pantalla se cerrará, guardando los datos del autor. Si por el contrario no queremos modificar ningún dato. Tenemos que clicar el botón de Cerrar (o las teclas “Alt” + “C”).

Nota: Si queremos consultar, todos los datos de nuestros autores, la mejor opción es usar el listado (4.1.4 Listar los autores). Pero si por el contrario, sólo queremos consultar los datos de una autor, podemos abrir la pantalla de “Modificar Autor” y cerrarla con el botón de Cerrar



Figura 48: Modificar Autor

4.1.3 Eliminar un autor

Si queremos eliminar un autor, tenemos que seleccionarlo y luego presionar el botón de eliminar. Nos aparecerá un mensaje como el siguiente:

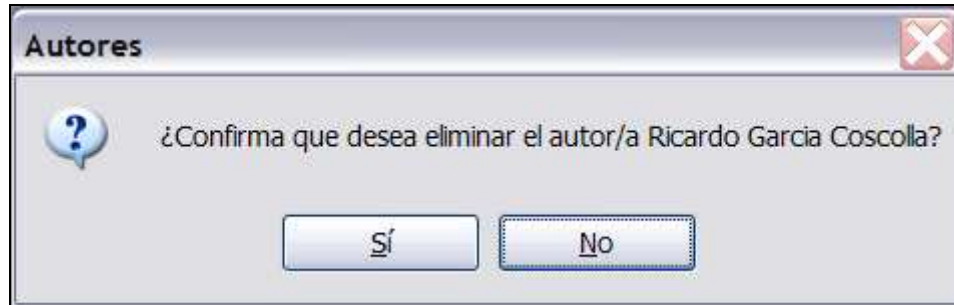


Figura 49: Eliminar un autor

Si el autor tiene libros publicados, no se puede eliminar y por lo tanto el programa nos avisará (ver figura 20). Si por el contrario el autor no tiene ningún libro publicado, este se eliminará y se mostrará un mensaje informativo en la barra de estado de la pantalla principal.

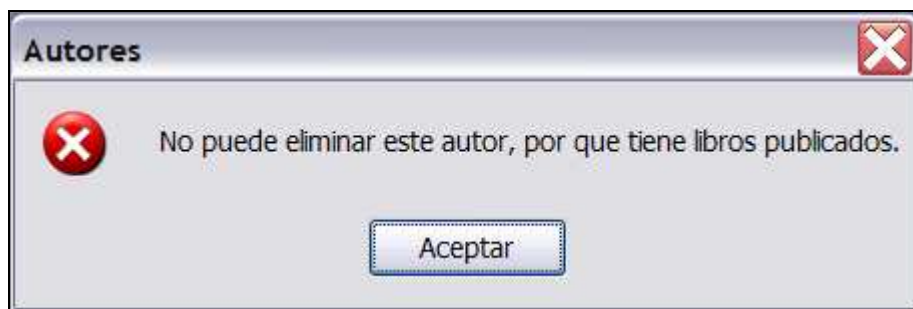


Figura 50: Error al eliminar un autor

4.1.4 Listar los autores

Si pulsamos el botón de imprimir, nos aparecerá (en otra ventana), el listado de autores. Con el podremos realizar varias operaciones:

- Visualizar los datos.
- Imprimirlos el listado en papel.
- Exportar el listado a un archivo Los formatos de los archivos, son estos: rpt, pdf, xls, doc y rtf.

También podemos consultar este listado desde el menú principal (Listados / Autores).

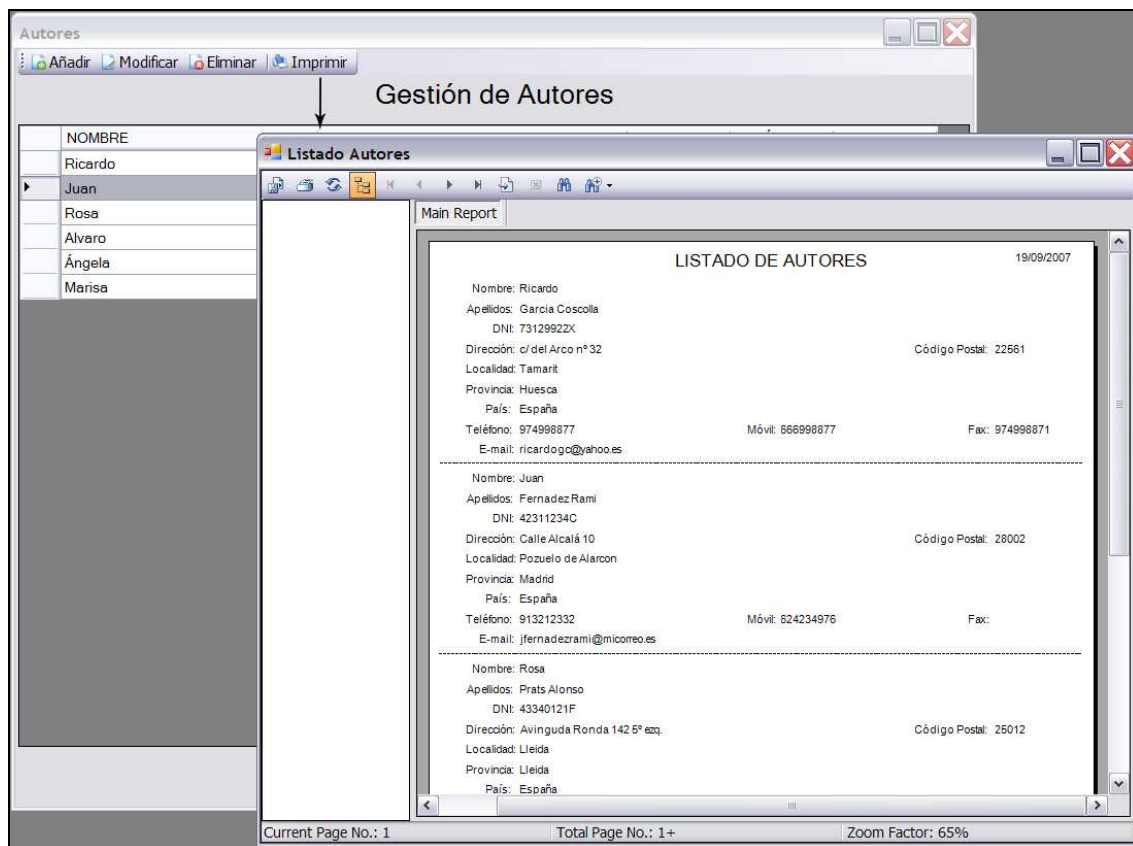


Figura 51: Listado de autores

4.2 Pantalla de Sellos, Estilos y Colecciones

Todos los libros pertenecen a una colección, esta a su vez pertenece a un estilo, y finalmente el estilo pertenece a un sello. Mediante esta jerarquía, el programa nos permite ordenar y clasificar nuestros libros. Esta funcionalidad está desarrollada en la pantalla de Sellos, Estilos y Colecciones (figura 22).

El funcionamiento es muy simple. Cuando seleccionamos un Sello, la lista de Estilos nos muestra los estilos de dicho sello. A su vez, cuando seleccionamos un nuevo estilo, la lista de colecciones mostrará sus colecciones. Por ejemplo (figura 22):

Los estilos del sello “Libros de Numismática S.A.” son: “Manuales de monedas” y “Manuales de billetes”. Al mismo tiempo, las colecciones de “Manuales de billetes”, son: “Billetes españoles”, “Billetes europeos” y “Billetes del mundo”.



Figura 52: Pantalla de sellos, estilos y colecciones.

La información de esta pantalla, se modificará con poca frecuencia. Es por ese motivo que no se muestra ningún botón (a excepción de botón de cerrar). Para Añadir, modificar o eliminar elementos en cada lista, tenemos que hacer un clic con el botón secundario del ratón en esta y nos aparecerá el siguiente menú:

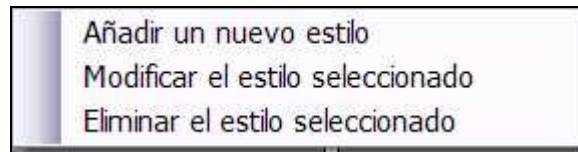


Figura 53: Menú contextual de la lista de estilos

4.2.1 Añadir un nuevo elemento a la lista

Como hemos explicado anteriormente, si queremos añadir un sello, estilo o colección, tenemos que presionar, el botón secundario del ratón, en la lista que queramos modificar. Luego seleccionaremos la opción de añadir. La aplicación mostrará una ventana como la siguiente:



Figura 54: Pantalla de nuevo Sello.

En esta pantalla, simplemente tenemos que introducir, el nombre del elemento (en el caso de la figura 24 será un sello), y presionar el botón “Aceptar” (o las teclas “Alt” + “A”).

4.2.2 Modificar un elemento

Para modificar un elemento, seleccionaremos la opción del menú “Modificar el elemento seleccionado” (en el ejemplo siguiente “Modificar la colección seleccionada”) y nos aparecerá una pantalla esta:



Figura 55: Pantalla de modificar colección.

Modificaremos lo que sea oportuno y pulsaremos el botón de Aceptar (o las teclas “Alt” + “A”). La ventana se cerrará guardando los cambios. Si no queremos cambiar nada, pulsaremos el botón de Cancelar (o las teclas “Alt” + “C”).

4.2.3 Eliminar un elemento

Eliminar un elemento (sello, estilo o colección), es una operación muy sencilla. Para ello, sólo tenemos que presionar en la opción “Eliminar el elemento seleccionado”, del menú emergente.

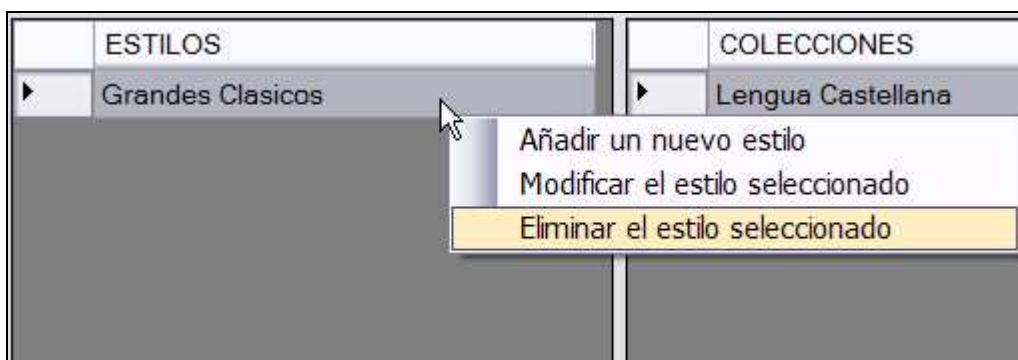


Figura 56: Eliminar un sello.

Si se puede eliminar el elemento, el sistema lo eliminará, así como sus dependencias. Pro ejemplo, si eliminamos un sello, se eliminará este y sus estilos y colecciones.

Por otra parte, si queremos eliminar un sello, estilo o colección con libros asociados a este, el sistema nos lo prohibirá y nos mostrará un mensaje de error.

4.3 Pantalla de Libros

Utilizaremos esta pantalla para hacer altas, modificaciones, bajas y listados de libros. Su funcionamiento es similar al del la pantalla de autores (revisar sección 4.1).

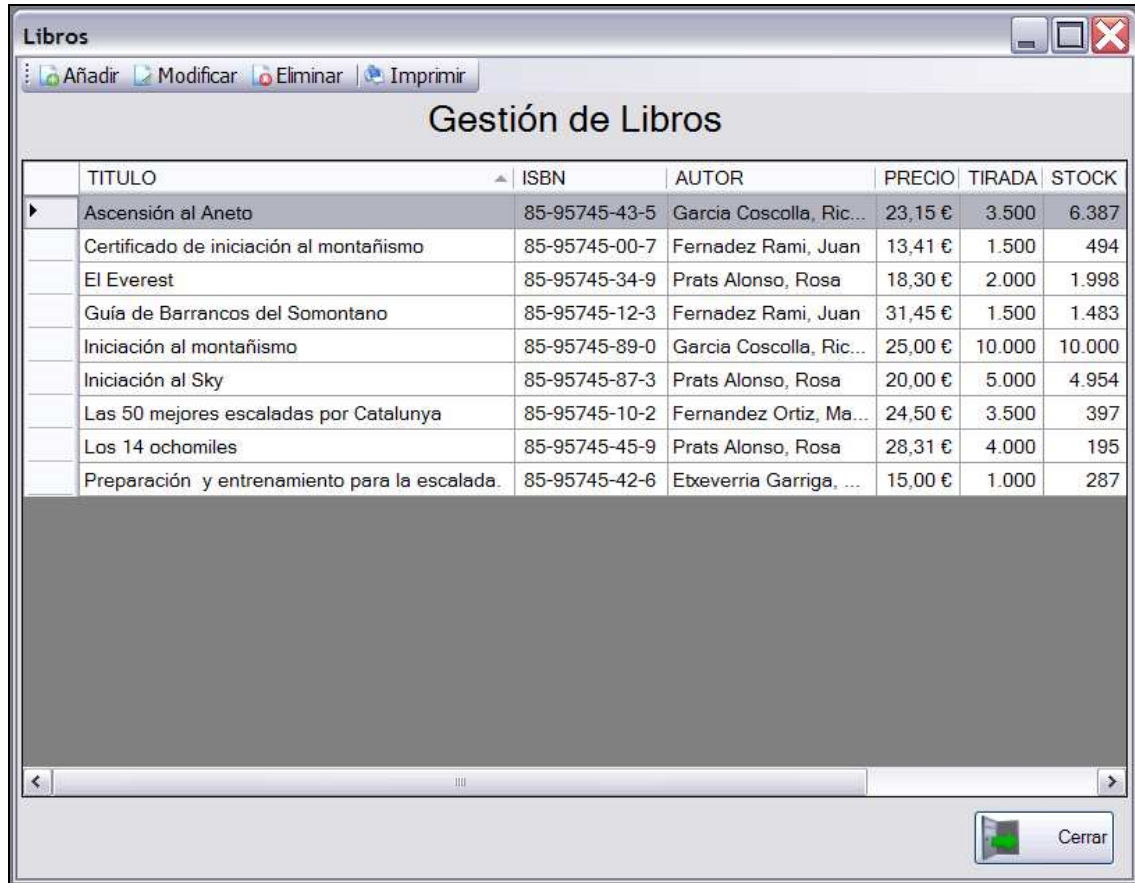


Figura 57: Pantalla de Libros

4.3.1 Añadir un libro

Añadir un libro nuevo al sistema, es una operación similar, a añadir un nuevo autor (revisar sección 4.2). Clicaremos el botón de nuevo y nos aparecerá la ficha vacía de este:

Figura 58: Nuevo Libro

Como podemos observar en la figura 28, la ventana tiene varias fichas:

- Datos Generales
- Datos Portada
- Datos Interior
- Datos Contraportada y Colores

Aunque la mayoría de estos datos, son opcionales (pueden quedarse en blanco), algunos son obligatorios como por ejemplo: el título, ISBN, Autor...

Otra peculiaridad de esta pantalla, son las cajas de texto de color gris. Este color indica, que la caja de texto, es de sólo lectura y por lo tanto, no la podemos editar. En estas cajas se nos muestra, información útil, relativa al libro.

Por último, podemos observar que al lado de la caja de texto del autor, hay un botón con una lupa. Si lo presionamos nos aparecerá una pantalla con todos los autores del sistema (ver Figura 29). Únicamente tendremos que seleccionar el autor del libro y pulsa Aceptar (o las teclas “Alt” + “A”).

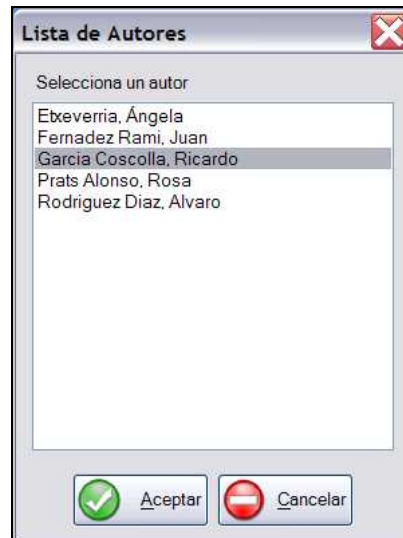


Figura 59: Lista de Autores

4.3.2 Modificar un libro

La modificación de los datos de un libro, se realiza de la misma forma que la modificación de los datos de un autor. Para modificar los datos de un libro, tenemos que acceder a su ficha. Cambiar lo que se desee. Y finalmente pulsar el botón de Guardar (“Alt” + “G”).

4.3.3 Eliminar un libro

Para eliminar un libro, primero tenemos que seleccionarlo y después tenemos que presionar el botón eliminar, de la barra de herramientas (en la pantalla de Libros).

Si el libro seleccionado, tiene movimientos (para consultar los movimientos Listados / Movimientos de libros), el sistema nos impedirá su eliminación y nos mostrará un mensaje de error.

4.3.4 Listar los libros

Para acceder al listado de libros, tenemos que pulsar el botón de listado (como se explica en el apartado 4.1.4). Una vez pulsado dicho botón, se nos abrirá una nueva pantalla (ver Figura 30). Este listado se podrá imprimir o exportar a pdf, xls, doc...

También podemos acceder al listado de libros el menú principal (Listados / Libros).

LISTADO DE LIBROS					10/08/2007
Sello / Estilo / Colección: GRAU Ed. / Guías de Aventura / Escalada					
ISBN	Título y autor	Tirada	Fecha P.	Precio	
12-34567-89-4	100 Escaladas en Catalunya <i>Rodriguez Diaz, Alvaro</i>	2.500	01/01/2007	25,00 €	
12-34567-89-5	Preparación para escaladas <i>Rodriguez Diaz, Alvaro</i>	1.200	10/08/2007	15,12 €	
Libros en GRAU Ed. / Guías de Aventura / Escalada : 2					
Sello / Estilo / Colección: GRAU Ed. / Guías de Aventura / Montaña					
ISBN	Título y autor	Tirada	Fecha P.	Precio	
12-34567-89-0	Iniciación al montañismo <i>Garcia Coscolla, Ricardo</i>	10.000	10/03/2006	25,00 €	
12-34567-89-1	Guía de Ordesa <i>Fernandez Rami, Juan</i>	4.002	04/07/2007	34,00 €	
12-34567-89-2	Anscensión al Everest <i>Garcia Coscolla, Ricardo</i>	20.000	01/08/2007	0,00 €	
12-34567-89-3	Preparación para expediciones <i>Prats Alonso, Rosa</i>	1.000	11/06/2007	15,00 €	
Libros en GRAU Ed. / Guías de Aventura / Montaña : 4					
Sello / Estilo / Colección: GRAU Ed. / Guías de Aventura / Rios y Barrancos					
ISBN	Título y autor	Tirada	Fecha P.	Precio	
12-34567-89-6	Guía de Barrancos <i>Etxeverria, Ángela</i>	1.500	21/10/2006	15,00 €	
12-34567-89-7	Guía de seguridad en Barrancos <i>Etxeverria, Ángela</i>	500	11/03/2007	12,50 €	
Libros en GRAU Ed. / Guías de Aventura / Rios y Barrancos: 2					
número total de libros: 8					

Figura 60: Listado de Libros

4.4 Pantalla de Almacén

La función de esta pantalla (ver figura 31), principalmente es la de informar del estado de nuestro almacén de libros. Para ello usamos dos listados “Todos los libros” y “Libros con stock inferior al Stock Min.”(Parte 2 de la pantalla). En el primero obtenemos información de todos los libros de nuestro almacén, mientras que en el segundo, sólo nos aparecerán los libros cuyo stock, sea inferior al stock mínimo del libro (este valor lo establecemos en las características del libro, ver Figura 28). Como podemos observar en nuestra figura 28, el libro “Iniciación al montañismo” tiene un stock de 9780 unidades.

Sin embargo, en esta pantalla también se realiza una operación imprescindible dentro del sistema, “la carga de libros al Almacén” (parte 1). Esta operación consiste en agregar nuevos ejemplares de un libro al almacén, para posteriormente poder abastecer a los distribuidores o vender a clientes.

ISBN	TITULO	Fecha P.	TIRADA	EDITADOS	STOCK
85-95745-89-0	Iniciación al montañismo	10/03/2006	10.000	10.000	9.750
85-95745-00-7	Certificado de iniciación al montañis...	06/06/2007	1.500	500	409
85-95745-10-2	Las 50 mejores escaladas por Catal...	12/09/2007	3.500	400	295
85-95745-12-3	Guía de Barrancos del Somontano	24/04/2007	1.500	1.500	1.140
85-95745-87-3	Iniciación al Sky	20/03/2007	5.000	5.000	4.635
85-95745-43-5	Ascensión al Aneto	01/02/2007	3.500	3.500	2.955
85-95745-42-6	Preparación y entrenamiento para l...	12/09/2007	1.000	300	160
85-95745-45-9	Los 14 ochomiles	13/11/2006	4.000	200	175
85-95745-34-9	El Everest	03/07/2007	2.000	2.000	1.430

Carga de libros al Almacén

Libro: Ascensión al Aneto

Cantidad:

Figura 61: Pantalla de Almacén

En la sección “Libros con stock inferior al Stock Min.”, nos aparecen los libros con stock inferior a su stock mínimo, por ejemplo en la figura 29 el libro “Guía de Ordesa” tiene 98 en stock, pero su stock mínimo es de 100 unidades. Si se cree conveniente editar nuevos ejemplares de dicho libro, se tendrá que realizar una “carga de libros al almacén” una vez que estos lleguen de la imprenta.

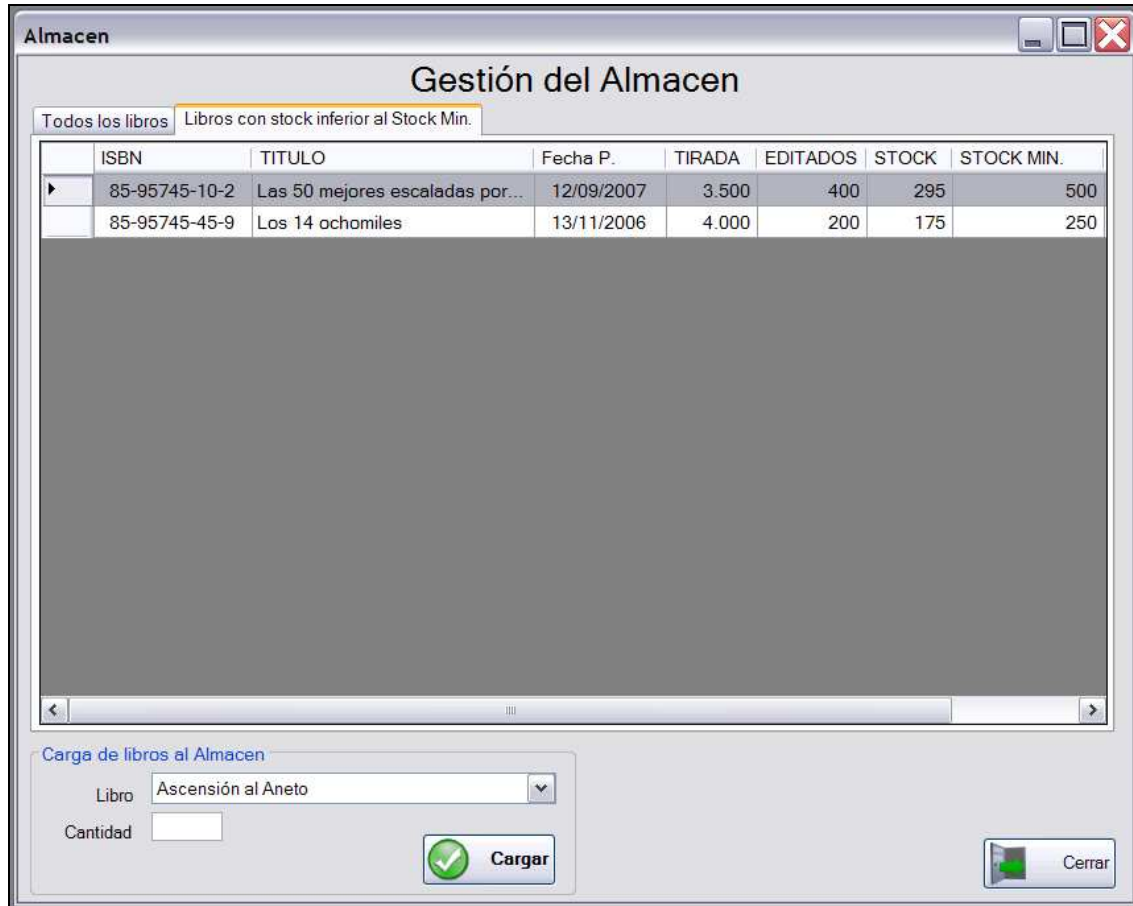


Figura 62: Pantalla de Almacén (Libros con Stock inferior al Stock Min.)

4.5 Pantalla de Clientes

Utilizaremos esta pantalla (ver figura 33), para hacer altas, modificaciones, bajas y listados de clientes. Su funcionamiento es similar al de la pantalla de autores (revisar sección 4.1). No se podrá eliminar un cliente, si tiene facturas.

Un cliente es una persona física o fiscal a la que únicamente le realizamos ventas de libros, o sea no es necesario tener un control de su stock. Esto es muy útil si tenemos que vender una cantidad pequeña de libros (como veremos en la sección de facturar a clientes).

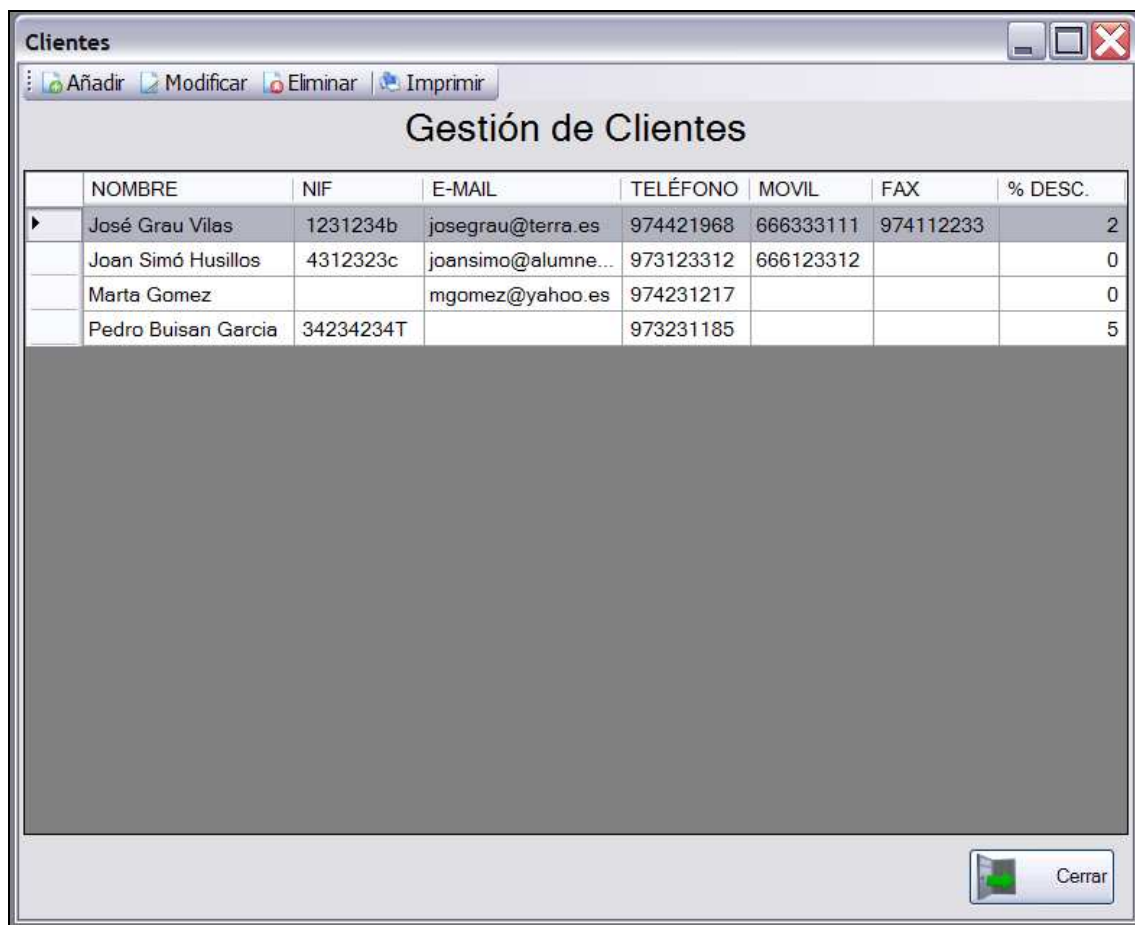


Figura 63: Pantalla Clientes

Para acceder al listado de Clientes, tenemos que pulsar el botón de imprimir (como se explica en el apartado 4.1.4). Una vez pulsado dicho botón, se nos abrirá una nueva pantalla con el listado (ver Figura 34). Este listado se podrá imprimir o exportar a pdf, xls, doc...

También podemos acceder al listado de libros el menú principal (Listados / Clientes).



Figura 64: Listado de Clientes

4.6 Pantalla de Distribuidores

Esta pantalla es la encargada de realizar altas, baja y modificaciones de los Distribuidores (ver figura 33). Su funcionamiento es el mismo que el de la pantalla de autores (revisar sección 4.1):

- Si pulsamos el botón de **añadir**, se nos abrirá una ficha vacía, solamente tenemos que poner los datos de distribuidor y presionar el botón de guardar (o las teclas “Alt” + “G”).
- Para eliminar distribuidor, primero tenemos que seleccionarlo, dentro de la lista de distribuidores y después presionaremos el botón de **eliminar**.
- Si queremos modificar los datos de un distribuidor, realizaremos doble clic en este (o bien, lo seleccionaremos y pulsaremos **modificar**), modificaremos los datos que deseemos en la ficha (ver figura 32) y pulsaremos el botón de guardar (o las teclas “Alt” + “G”).



Modificar Distribuidor

Nombre: La Casa del Libro

Datos Generales Datos Adicionales

Nombre La Casa del Libro

NIF 123123123

Dirección Av. Gran Vía nº 122

Código Postal 28007

Localidad Madrid

Provincia Madrid

País España

Teléfono 91123123

T. Movil

Fax 91123123

E-mail tienda@casalibro.com

Guardar Cerrar

Figura 65: Ficha de un Distribuidor

Desde esta pantalla, podemos observar el stock que tiene cada distribuidor, a sí como el stock de nuestro almacén (operación que también podemos realizar desde la pantalla “Gestión del almacén”).

Esta pantalla, como todas las anteriores, se puede maximizar par consultar mas cómodamente su contenido, pero también podemos **redimensionar los listados**, por si nos interesa aumentar el tamaño al listado de libros o al de distribuidores.

Distribuidores

Añadir Modificar Eliminar Imprimir

Gestión de Distribuidores

	NOMBRE	NIF	E-MAIL	TELÉFONO	MOVIL	FAX	DESC.
	Mi Editorial	73204825-X	info@miempresa.com	974421919	674421919	974421919	0
▶	La Casa del Libro	09856345Y	tienda@casalibro.com	91123123		91123123	12
	La Casa de las Novelas	74939202H	info@lacasadelasnov...	974224824		974224824	12
	Katya Librería	23034934R	lib-katya@terra.es	974428652		974428652	0
	Librería Edelweis	94593453D		974226650			5
	Llibreria Caselles	23942934F		973123344	673123344	973123344	15

Listado le libros en stock, del distribuidor seleccionado:

	ISBN	TITULO	PVP sin IVA	STOCK
▶	85-95745-89-0	Iniciación al montañismo	25,00 €	250
	85-95745-87-3	Iniciación al Sky	20,00 €	50
	85-95745-00-7	Certificado de iniciación al montañismo	13,41 €	15
	85-95745-10-2	Las 50 mejores escaladas por Catalunya	24,50 €	35
	85-95745-12-3	Guía de Barrancos del Somontano	31,45 €	150
	85-95745-43-5	Ascensión al Aneto	23,15 €	300
	85-95745-42-6	Preparación y entrenamiento para la escalada.	15,00 €	50
	85-95745-34-9	El Everest	18,30 €	500

Cerrar

Figura 66: Pantalla de Distribuidores

Para acceder al listado de Distribuidores, tenemos que pulsar el botón de imprimir (como se explica en el apartado 4.1.4). Una vez pulsado dicho botón, se nos abrirá una nueva pantalla con el listado (ver Figura 37). Este listado se podrá imprimir o exportar a pdf, xls, doc...

También podemos acceder al listado de libros el menú principal (Listados / Distribuidores).

LISTADO DE DISTRIBUIDORES		04/09/2007
Nombre: La Casa del Libro		
DNI: 123123123		
Dirección: Av. Gran Vía nº 122		Código Postal: 28007
Localidad: Madrid		
Provincia: Madrid		
País: España		
Teléfono: 91123123		Móvil: Fax: 91123123
E-mail: tienda@casalibro.com		
Descuento: 12%		

Figura 67: Listado de Distribuidores

4.7 Pantalla de Movimientos

Esta pantalla consta de dos pestañas “Ventas y Movimientos entre distribuidores” (ver figura 38) y “Rectificación de una venta” (ver figura 40).

Movimientos

Movimientos entre Distribuidores

Ventas y Movimientos entre distribuidores Rectificación de una venta

Elige una opción...

☒ Movimiento entre distribuidores.

☐ Venta.

☐ Venta directa.

Información rápida:

Para realizar un movimiento de libros, entre distribuidores. Primero seleccione el distribuidor origen. Luego, seleccione un libro de su stock. Introduzca la cantidad. Y por último, seleccione el distribuidor destino.

1. Distribuidor: Mi Editorial

2. Libros en stock

	ISBN	TITULO	STOCK
▶	85-95745-89-0	Iniciación al montañismo	9.750
	85-95745-00-7	Certificado de iniciación al montañismo	409
	85-95745-10-2	Las 50 mejores escaladas por Catalunya	295
	85-95745-12-3	Guía de Barrancos del Somontano	1.140
	85-95745-87-3	Iniciación al Sky	4.635
	85-95745-43-5	Ascensión al Aneto	2.955
	85-95745-42-6	Preparación y entrenamiento para la escalada.	160
	85-95745-45-9	Los 14 ochomiles	175
	85-95745-34-9	El Everest	1.430

3. Cantidad:

4. Distribuidor Destino: Katya Librería

☒ Confirmar

Figura 68: Pantalla Movimientos (Pestaña “Ventas y Movimientos”)

4.7.1 Pestaña “Ventas y Movimientos entre distribuidores”

La pestaña “Ventas y Movimientos entre distribuidores”, consta de dos partes:

1. “Elige una opción”, establecemos que operación queremos realizar.

Operaciones:

 - a. **Movimiento entre distribuidores** esta opción no sirve para mover una cantidad X de libros de un distribuidor a otro, o desde nuestro almacén (observando la figura 35, en nuestro caso “Mi Editorial”) a un distribuidor. Una vez realizada esta operación, observaremos que los stocks se actualizan (esto se puede ver desde esta pantalla o la pantalla de distribuidores).
 - b. **Venta**, utilizamos esta opción, para establecer que un distribuidor, a realizado una venta de X unidades, de un determinado libro. Por lo tanto su stock se actualizará y podremos posteriormente facturar este movimiento.

Movimientos

Movimientos entre Distribuidores

Ventas y Movimientos entre distribuidores Rectificación de una venta

Elige una opción...

☐ Movimiento entre distribuidores.

☒ Venta.

☐ Venta directa.

Información rápida:
Para realizar una venta. Primero seleccione el distribuidor. Luego, seleccione un libro de su stock. Y por último, introduzca la cantidad que ha vendido.

1. Distribuidor **La Casa de las Novelas**

2. Libros en stock

	ISBN	TITULO	STOCK
▶	85-95745-00-7	Certificado de iniciación al montañismo	25
	85-95745-10-2	Las 50 mejores escaladas por Catalunya	7
	85-95745-12-3	Guía de Barrancos del Somontano	75
	85-95745-87-3	Iniciación al Sky	260
	85-95745-43-5	Ascensión al Aneto	125
	85-95745-42-6	Preparación y entrenamiento para la escalada.	6
	85-95745-45-9	Los 14 ochomiles	5
	85-95745-34-9	El Everest	25

3. Cantidad

☒ Confirmar

Figura 69: Pantalla Movimientos (Opción “Venta”)

- c. **Venta directa**, esta opción combina las dos opciones anteriores, o sea, al mismo tiempo se está realizando un movimiento de X ejemplares de un libro y el distribuidor destino los vende. El resultado es una disminución de X ejemplares del libro seleccionado, en el stock del distribuidor origen y X ejemplares de dicho libro, pendientes para facturar en el distribuidor destino.
2. En esta sección estableceremos el distribuidor, el libro, la cantidad... El contenido cambiará si realizamos una venta (ver figura 39).

4.7.2 Pestaña “Rectificación de una venta”

Esta pestaña, es de gran utilidad, si se ha producido un error, a la hora de establecer una venta. Siempre y cuando no hayamos generado la factura podremos realizar un abono de esa venta. Por ejemplo, si queríamos vender 4 ejemplares, en lugar de 5, seleccionemos el libro y podremos 1 en la cantidad, de esta forma el sistema actualizará el stock (del distribuidor que havia realizado la venta) y la los ejemplares vendidos serán 4.

Esta opción es importante si se realizan devoluciones en el distribuidor que ha efectuado la venta.

Movimientos

Movimientos entre Distribuidores

Ventas y Movimientos entre distribuidores Rectificación de una venta

Información rápida:
Para realizar una rectificación, de una venta. Primero seleccionaremos el distribuidor que realizó la venta. Una vez hecho esto, veremos todas sus ventas pendientes de facturar. Seleccionaremos aquella que deseamos corregir. Por último, pondremos la cantidad que queremos restar a las unidades vendidas.
Por ejemplo, si el distribuidor X, vendió 500 unidades del libro L, en lugar de 750. En el campo cantidad pondremos 250.

1. Distribuidor:

2. Ventas pendientes de facturar

	ISBN	TITULO	CANTIDAD	PRECIO
▶	85-95745-12-3	Guía de Barrancos del Somontano	7	31,45 €

3. Cantidad:

Figura 70: Pantalla Movimientos (Pestaña “Rectificación de una venta”)

4.8 Pantalla de Facturas

Esta Pantalla es una de las más importantes de la aplicación y consta de las siguientes pestañas:

4.8.1 Pestaña Listado de facturas

Desde pestaña esta podemos consultar todas las facturas, ordenarlas, imprimirlas o establecer como cobrada una factura no cobrada.

Como podemos observar en el la figura 41, el listado sólo nos muestra los datos más importantes de la factura, si queremos verla entera, tendremos que realizar doble clic en ella, o pulsar el botón de Imprimir.

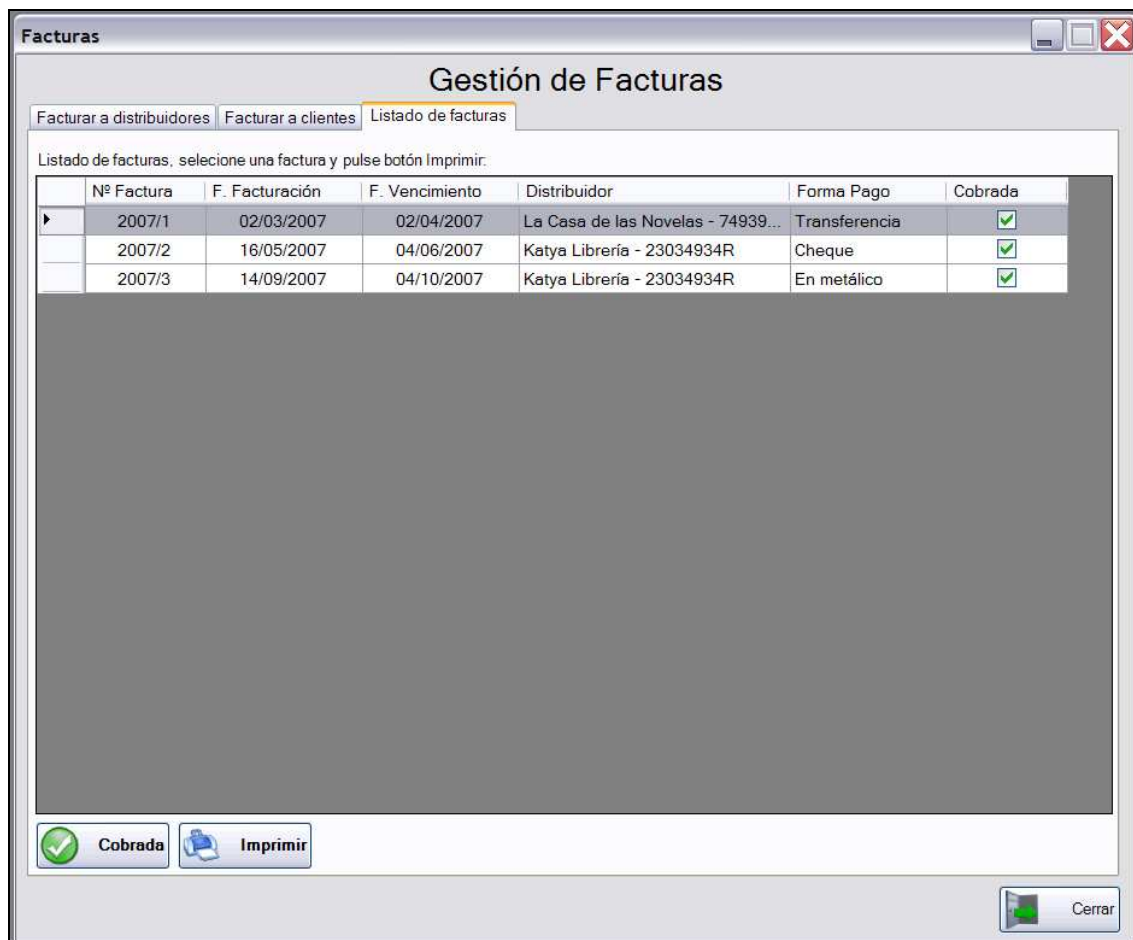


Figura 71: Pantalla Facturas

En la parte inferior de la pantalla, encontramos dos botones. Uno de ellos nos sirve para establecer una factura como cobrada (antes el sistema nos preguntara si deseamos realizar dicha operación). Y el otro para imprimir una o varias facturas. Si pulsamos el botón de imprimir nos aparece la siguiente pantalla:

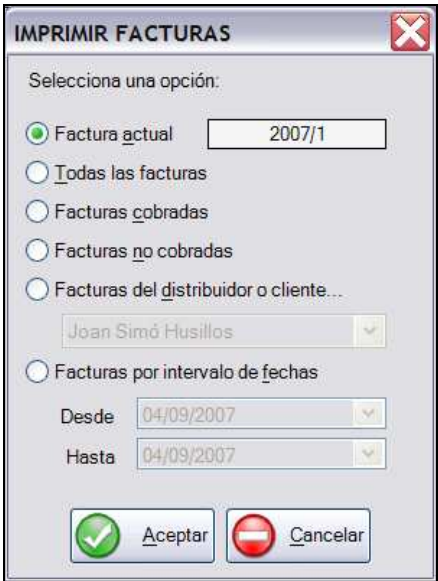


Figura 72: Pantalla Imprimir Facturas

Como podemos observar tenemos hasta 6 tipos de filtros diferentes, para visualizar y/o imprimir las facturas deseadas. En este ejemplo (ver figura 42), sólo se nos mostrará la factura 2007/1 (ver figura 43).

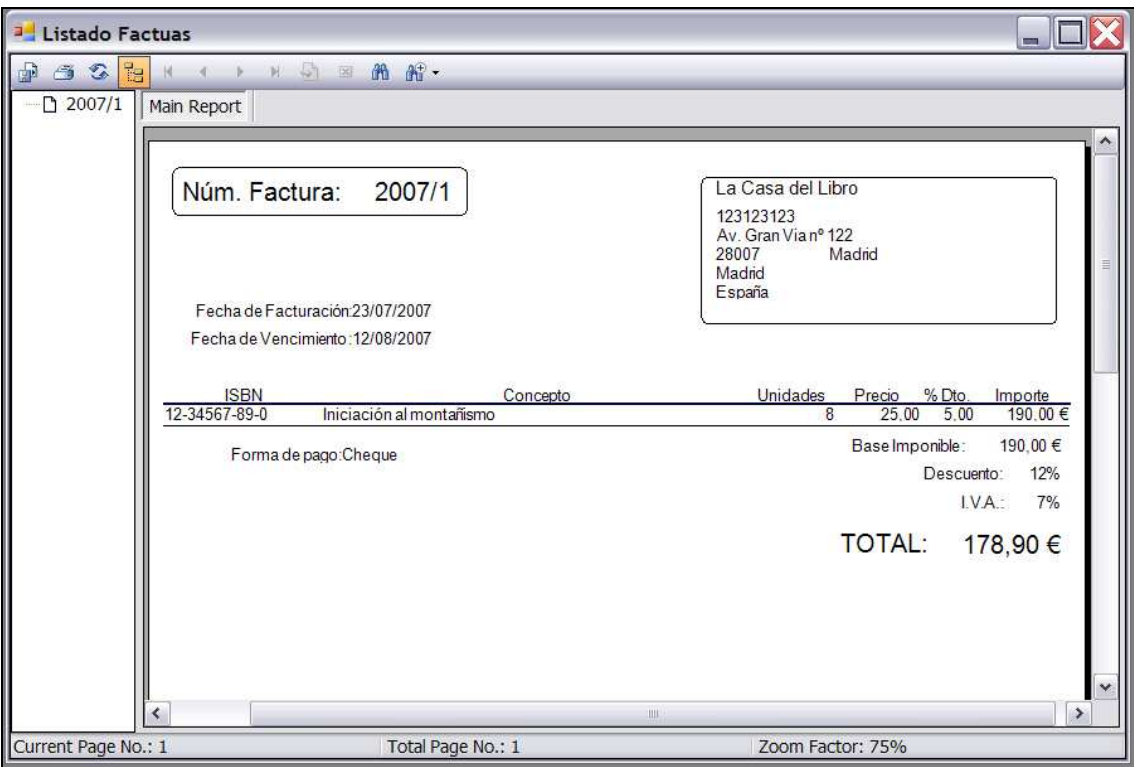


Figura 73: Listado Facturas

4.8.2 Pestaña Facturar a Distribuidores

Como hemos explicado anteriormente, la pantalla de facturas consta de 3 pestañas, en la primera de ellas podremos generar facturas a distribuidores, facturando las ventas pendientes.

Figura 74: Pestaña Facturar a Distribuidores

En la pestaña de facturar a distribuidores (ver figura 44), lo primero que tenemos que hacer es seleccionar el distribuidor. Una vez hecho esto, en la lista inferior-izquierda, nos aparecerán sus libros pendientes de facturar.

Posteriormente tenemos que establecer la fecha de facturación y vencimiento, el IVA y la forma de pago (sin especificar, transferencia, t. crédito, en metálico y cheque).

Luego seleccionamos las líneas (con la ayuda de los cuatro botones de selección), que deseamos facturar. Y pulsamos el botón Generar (o las teclas “Atl” + “G”).

Finalmente nos aparece una vista preliminar de la factura:

FACTURA N° 2007/4

Núm. Factura: 2007/4

Datos de la Factura

Fecha Facturación: 18/09/2007
Fecha Vencimiento: 08/10/2007
IVA: 7 %
Forma de Pago: Transferencia

Factura no cobrada

Datos Cliente / Distribuidor

Nombre: La Casa del Libro
Dirección: Av. Gran Vía nº 122
N.I.F.: 09856345Y
Localidad: Madrid
C.P.: 28007
Provincia: Madrid
País: España

	ISBN	CONCEPTO	% DESC.	UNIDADES	PRECIO	SUBTOTAL
▶	85-95745-12-3	Guía de Barrancos del Somontano	0	7	31,45 €	220,15 €
	85-95745-43-5	Ascensión al Aneto	0	20	23,15 €	463,00 €
	85-95745-34-9	El Everest	0	50	18,30 €	915,00 €
	85-95745-87-3	Iniciación al Sky	0	10	20,00 €	200,00 €

Información rápida:
Esto es una vista previa de la factura. Por favor, revisela con atención y pulse Aceptar si esta de acuerdo. En caso contrario, pulse Cancelar.

Base Imponible..... 1798,15 €
% Dto. Distribuidor..... 12
% I.V.A. 7

Total..... 1693,14 €

✓ Aceptar

✗ Cancelar

Figura 75: Pantalla Vista preliminar de la Factura

Si estamos de acuerdo pulsaremos el botón aceptar (o las teclas “Atl” + “A”). En caso contrario, pulsaremos el botón cancelar (o las teclas “Atl” + “C”).

4.8.3 Pestaña Facturar a Clientes

Esta es la pestaña (ver figura 46) para facturar a clientes. Una vez realizados todos los pasos, si le clicamos el botón generar se generará una nueva factura, a nombre del cliente seleccionado. Y los libros facturados se restan, del stock del almacén.

Facturas

Gestión de Facturas

Facturar a distribuidores | **Facturar a clientes** | Listado de facturas

Datos de la Factura

Fecha Facturación: 19/09/2007
Fecha Vencimiento: 08/10/2007
IVA: 7 %
Forma de Pago: En metálico
☐ Cobrada

Datos Cliente

Nombre: Pedro Buisan Garcia
Dirección: Av. Blondel 29 3er 1ª
N.I.F.: 34234234T
C.P.: 25002 Localidad: Lleida
Provincia: Lleida País: España

Agregar nueva línea a la factura

Libro: Las 50 mejores escaladas por Catalunya
ISBN: 85-95745-10-2 PVP sin IVA: 24,50 €
Unidades: Stock: 295

ISBN	TÍTULO	UNIDADES	PRECIO
85-95745-89-0	Iniciación al montañismo	1	25,00 €

Figura 76: Pantalla Facturar a Clientes

El funcionamiento de esta pantalla es muy simple. Se puede resumir en los siguientes pasos:


1. Lo primero que tenemos que hacer, es **seleccionar un cliente** de la lista (dentro del grupo “Datos Cliente”), una vez hecho esto, podremos observar como se actualizan los campos dirección, N.I.F, C.P., localidad, provincia y país.
2. Posteriormente introduciremos los “**Datos de la Factura**” (dentro del grupo con el mismo nombre), estos son: fecha facturación, fecha vencimiento, IVA, forma pago y finalmente si está cobrada o no.

3. Una vez hemos hecho los dos pasos anteriores, es el momento de **añadir líneas de detalle** a nuestra factura. Para ello tenemos que realizar los siguientes pasos:
 - a. Seleccionar un libro de la lista, una vez hecho esto nos aparecerá por pantalla su ISBN, su precio sin IVA y lo que es más importante su stock en el almacén.
 - b. Una vez seleccionado el libro, es el momento de poner las unidades que deseamos vender de dicho libro. Este valor, tiene que ser inferior al stock, en caso contrario el programa nos mostrará un mensaje de error.
 - c. Por último, pulsaremos el botón Añadir (o las teclas “Alt.” + “ñ”), para que los datos se añadan a las líneas detalle de la factura.
 - d. Si queremos **eliminar una línea** de nuestra factura, tenemos que seleccionarla y clicar el botón de Quitar (o las teclas “Alt.” + “q”).
4. Por último, una vez introducidas todas las líneas de nuestra factura **pulsaremos generar** (o las teclas “Alt” + “G”), la aplicación nos mostrará una vista preliminar (ver figura 45), y nosotros la aceptaremos si estamos de acuerdo, o pulsaremos cancelar en caso contrario.

5. Otros Listados

En este punto explicaremos el funcionamiento de los listados que no han sido explicados en el punto 4, por no estar asociados a ninguna pantalla. Estos listados (junto con los anteriores), los encontramos en menú principal de la aplicación:

- Listado Clientes con sus ventas.
- Listado Distribuidores con su stock.
- Listado del stock del almacén.
- Listado de movimientos. Como este listado podemos aplicar los filtros que se muestran en la figura 47.
- Listado de Todas las facturas.



IMPRIMIR MOVIMIENTOS

Selecciona una opción:

☒ Todos los movimientos

☐ Movimientos del libro...

Ascensión al Aneto

☐ Movimientos del distribuidor...

Katya Librería

☐ Movimientos por intervalo de fechas

Desde 19/09/2007

Hasta 19/09/2007

Aceptar Cancelar

Figura 77: Imprimir movimientos.

ANEXO II: Clase Basedatos

La clase Basedatos, es un clase importantísima dentro de la implementación de la aplicación. Es el canal de conexión entre la base de datos y los formularios. A continuación se muestra el código fuente de esta:

```
Imports System.Data.SqlClient

''' <summary>
''' La clase Basedatos es la encargada de acceder a los datos a traves
de ADO.NET (usando SQL server manager)
''' </summary>
''' <remarks></remarks>
Public Class Basedatos
    Implements IDisposable

#Region "Variables miembro"
    ''' <summary>
    ''' Variable miembro donde almacenaremos la conexión a la BD.
    ''' </summary>
    ''' <remarks></remarks>
    Private conexion As SqlConnection
#End Region

#Region "Métodos privados"
    ''' <summary>
    ''' Método privado para abrir la conexión al la BD.
    ''' </summary>
    ''' <remarks></remarks>
    Private Sub Abrir()
        If conexion Is Nothing Then conexion = New
SqlConnection(My.Resources.CandenaConex)
        conexion.Open()
    End Sub
    ''' <summary>
    ''' Método público para cerrar la conexión al la BD.
    ''' </summary>
    ''' <remarks></remarks>
    Private Sub Cerrar()
        If conexion IsNot Nothing Then conexion.Close()
    End Sub
#End Region

#Region "Métodos y Funciones públicas"
    ''' <summary>
    ''' Función que nos devuelve datos de la BD.
    ''' </summary>
    ''' <param name="TipoComando">Variable de tipo enum, donde
especificamos el tipo de consulta
    ''' (text = consulta SQL, StoredProcedure = Procedimiento
almacenado, TableDirect = Tabla de la BD).</param>
    ''' <param name="ProcOConsultaSQL">Nombre del procedimiento
almacenado o tabla. O consulta SQL.</param>
    ''' <param name="NombreTabla">Nombre de la tabla que
retonaremos.</param>
    ''' <param name="Parametros">Lista de paramatros (Sólo cuando
TipoComando es un StoredProcedure)</param>
    ''' <returns>Un objeto contenedor de datos de tipo
DataTable.</returns>
    ''' <remarks></remarks>
```



```
Public Function ObtenerDatos(ByVal TipoComando As CommandType,
ByVal ProcOConsultaSQL As String, Optional ByVal NombreTabla As String
= Nothing, Optional ByVal Parametros As List(Of SqlParameter) =
Nothing) As DataTable
    Dim comando As SqlCommand
    Dim adaptador As New SqlDataAdapter() 'Creamos un objeto de
tipo SqlDataAdapter para llenar el DataTable.
    Dim dt As New DataTable() 'Creamos un objeto de tipo DataSet,
que representa una base de datos en memoria.

    Me.Abrir() 'abrimos la connexion con la BD.

    comando = New SqlCommand(ProcOConsultaSQL, conexion)
    comando.CommandType = TipoComando

    'Operador AndAlso. Realiza una conjunción lógica
"cortocircuitada" en dos expresiones.
    'Si (Parametros IsNot Nothing) FALSE => (Parametros.Count > 0)
no se evalúa
    '
    If TipoComando = CommandType.StoredProcedure AndAlso
Parametros IsNot Nothing AndAlso Parametros.Count > 0 Then
        For Each parametro As SqlParameter In Parametros
            comando.Parameters.Add(parametro)
        Next
    End If

    adaptador.SelectCommand = comando
    adaptador.Fill(dt)

    comando.Dispose()
    Me.Cerrar() 'cerramos la connexion con la BD.

    If NombreTabla IsNot Nothing Then dt.TableName = NombreTabla
    'Cambiamos el nombre de la tabla.
    Return dt
End Function
''' <summary>
''' Función que nos devuelve datos de la BD.
''' </summary>
''' <param name="ConsultaSQL">consulta SQL.</param>
''' <returns>Una instancia de
System.Data.SqlClient.SqlDataReader</returns>
''' <remarks>Una vez usado del reader es <b>hay que cerrar la
conexion a la BD,</b> usando el método CerrarConexion()</remarks>
Public Function ObtenerDatos(ByVal ConsultaSQL As String) As
SqlDataReader
    Dim reader As SqlDataReader
    Dim comando As SqlCommand

    Me.Abrir() 'abrimos la connexion con la BD.

    comando = New SqlCommand(ConsultaSQL, conexion)
    comando.CommandType = CommandType.Text

    reader = comando.ExecuteReader()

    comando.Dispose()
    'Me.Cerrar() 'cerramos la connexion con la BD.

    Return reader
```

```

End Function
''' <summary>
''' Función que nos crea una copia de seguridad de la BD.
''' </summary>
''' <param name="ruta">La ruta y el fichero donde se creará la
copia de seguridad.</param>
''' <returns>True si OK, en caso contrario, False.</returns>
''' <remarks></remarks>
Public Function Backup(ByVal ruta As String) As Boolean
    Dim cmdBackUp As SqlCommand
    Dim sBackup As String = "BACKUP DATABASE [" &
My.Resources.rutaBaseDatos & "] TO DISK = N'" & ruta _
        & "' WITH NOFORMAT, NOINIT, NAME =
N'C:\GESTEDI\GESTEDI.MDF-Completa Base de datos Copia de seguridad', "
    _
        & "SKIP, NOREWIND, NOUNLOAD, STATS =
10"

    Abrir() 'abrimos la connexion con la BD.
    Try
        cmdBackUp = New SqlCommand(sBackup, conexion)
        cmdBackUp.ExecuteNonQuery()

        Return True
    Cerrar() 'cerramos la connexion con la BD.
    Catch ex As Exception

        Cerrar() 'cerramos la connexion con la BD.
        Return False
    End Try
End Function
''' <summary>
''' Función que nos reatura la BD, desde una copia de seguridad.
''' </summary>
''' <param name="ruta">La ruta y el fichero desde donde se
restaurará la BD.</param>
''' <returns>True si OK, en caso contrario, False.</returns>
''' <remarks></remarks>
Public Function Retaurar(ByVal ruta As String) As Boolean
    Dim cmdBackUp As SqlCommand
    Dim sBackup As String = "RESTORE DATABASE [" &
My.Resources.rutaBaseDatos & "] FROM DISK = N'" & ruta _
        & "' WITH FILE = 1, NOUNLOAD, STATS
= 10"

    Dim csb As New SqlConnectionStringBuilder
    csb.DataSource = "INSPIRON-6400\SQLEXPRESS"
    csb.AttachDBFilename = My.Resources.rutaBaseDatos
    'csb.UserID = "INSPIRON-6400\Sergio"
    csb.IntegratedSecurity = True

    Using con As New SqlConnection(csb.ConnectionString)

        Try
            con.Open()
            cmdBackUp = New SqlCommand(sBackup, con)
            cmdBackUp.ExecuteNonQuery()

            con.Close()
            Return True
        Catch ex As Exception

```

```

        MessageBox.Show(ex.Message)
        Return False
    End Try
End Using
End Function
''' <summary>
''' Función que nos devuelve un código.
''' </summary>
''' <param name="consultaSQL">Variable donde le pasamos la
consulta SQL.</param>
''' <returns></returns>
''' <remarks></remarks>
Public Function ObtenerCodigo(ByVal consultaSQL As String) As
Integer
    Dim codigo As Integer = 0
    Dim comando As SqlCommand

    Abrir() 'abrimos la connexion con la BD.

    comando = New SqlCommand(consultaSQL, conexion)
    comando.CommandType = CommandType.Text

    Try
        codigo = comando.ExecuteScalar()
    Catch ex As Exception
        codigo = 0
    End Try

    comando = Nothing
    Cerrar()

    Return codigo
End Function
''' <summary>
''' Método que ejecuta un procedimiento almacenado en la BD.
''' </summary>
''' <param name="NombreProc">Nombre del prodimento
almacendo</param>
''' <param name="parametros">Lista de paramatros</param>
''' <returns>devuelve 1 si ok.</returns>
''' <remarks></remarks>
Public Function EjecutarProcedimiento(ByVal NombreProc As String,
Optional ByVal parametros As List(Of SqlParameter) = Nothing) As
Integer
    Dim comando As SqlCommand
    Abrir() 'Abrimos la connexion con la BD.

    comando = New SqlCommand(NombreProc, conexion)
    comando.CommandType = CommandType.StoredProcedure

    If parametros IsNot Nothing OrElse parametros.Count > 0 Then
        For Each parametro As SqlParameter In parametros
            comando.Parameters.Add(parametro)
        Next
    End If

    comando.Parameters.Add(New SqlParameter("ReturnValue",
SqlDbType.Int, 4, ParameterDirection.ReturnValue, False, 0, 0,
String.Empty, DataRowVersion.Default, Nothing))

    comando.ExecuteNonQuery()

```

```

        Cerrar() 'Cerramos la connexion con la BD.

        Return comando.Parameters("ReturnValue").Value
    End Function
    ''' <summary>
    ''' Función para crear un paramtro del tipo SqlParameter.
    ''' </summary>
    ''' <param name="NombreParametro">Nombre del parametro.</param>
    ''' <param name="TipoBD">Tipo de variable SQL server.</param>
    ''' <param name="Tamano">Tamaño de la variable.</param>
    ''' <param name="ParametroEntrada">True = parametro entrada, False
= parametro salida.</param>
    ''' <param name="Valor">Si es un paramtro de entrada, le pasamos
el valor.</param>
    ''' <returns>un objeto SqlParameter</returns>
    ''' <remarks></remarks>
    Public Function CrearParametro(ByVal NombreParametro As String,
ByVal TipoBD As SqlDbType, ByVal Tamano As Integer, _
    ByVal ParametroEntrada As Boolean, Optional ByVal Valor As Object
= Nothing) As SqlParameter
        Dim Parametro As SqlParameter

        If Tamano > 0 Then
            Parametro = New SqlParameter(NombreParametro, TipoBD,
Tamano)
        Else
            Parametro = New SqlParameter(NombreParametro, TipoBD)
        End If

        If ParametroEntrada = True Then
            Parametro.Direction = ParameterDirection.Input
            If Valor IsNot Nothing Then Parametro.Value = Valor
        Else
            Parametro.Direction = ParameterDirection.Output
        End If

        Return Parametro
    End Function
    ''' <summary>
    ''' Método público para cerrar la conexión al la BD.
    ''' </summary>
    ''' <remarks></remarks>
    Public Sub CerrarConexion()
        If conexion IsNot Nothing Then conexion.Close()
    End Sub
    ''' <summary>
    ''' Implementamos el métodos Dispose de la interficie IDisposable.
    ''' </summary>
    ''' <remarks></remarks>
    Public Sub Dispose() Implements System.IDisposable.Dispose
        'Nos aseguramos que la conexión este cerrada.
        If conexion IsNot Nothing Then
            conexion.Dispose()
            conexion = Nothing
        End If
    End Sub
#End Region
End Class

```